

DEVELOPMENT OF NOVEL TECHNIQUES TO STUDY NONLINEAR ACTIVE NOISE CONTROL

*Thesis submitted in partial fulfillment
of the requirements for the degree of*

Doctor of Philosophy

By

Kunal Kumar Das

Roll No. 50602001

Under the Supervision of
Prof. Jitendriya Ku. Satapathy



Department of Electrical Engineering
National Institute of Technology
Rourkela, INDIA
2011

... dedicated to my loving parents

CERTIFICATE

This is to certify that the thesis entitled “**Development of Novel Techniques to Study Nonlinear Active Noise Control**”, submitted to the National Institute of Technology, Rourkela (INDIA) by **Kunal Kumar Das**, Roll No. 50602001 for the award of the degree of Doctor of Philosophy in Electrical Engineering, is a bonafide record of research work carried out by him under my supervision and guidance. The thesis, which is based on candidate’s own work, has not been submitted elsewhere for a degree/diploma. This thesis fulfills all the prescribed requirements for the award of Doctor of Philosophy.

Dr. J. K. Satapathy, Ph. D. (Bradford)

Vice Chancellor

BPUT, ROURKELA

(on lien from

Department of Electrical Engineering

National Institute of Technology

ROURKELA)

ACKNOWLEDGEMENTS

The work presented in this dissertation would not have been possible without the help and support of a large number of people. I first express my heartiest gratitude to my guide and supervisor **Prof. J.K. Satapathy**, Vice Chancellor, Biju Patnaik University of Technology, Rourkela (on lien from Department of Electrical Engineering, National Institute of Technology, Rourkela) for his valuable guidance, inspiration and encouragement in the course of the present work. The successful completion of the work is due to his constant inspiration and motivation. Working with my supervisor is highly enjoyable, inspiring and rewarding experience.

I humbly acknowledge the creative suggestions and constructive criticism of Prof Bidyadhar Subudhi, HOD, Electrical Engineering, Prof. Sarat Kumar Patra and Prof. Susmita Das committee members, while scrutinizing my research results. I express my sincere thanks to Prof. Pradeepta Kumar Nanda for his valuable comments and advice and also faculty members of the Department of Electrical Engineering, NIT Rourkela for constant advice, useful discussions and encouragement in pursuing the research work.

The help and cooperation received from the Dean and Head of the Department of Electronics and Communication Engineering, ITER, Bhubaneswar are gratefully acknowledged.

Last but not the least I expresses my special thanks to my parents, wife(Ruby), daughter(Adya), sisters and brothers for their prayer and all the inspiration for the last six years for successful completion of this research work.

Kunal Kumar Das

CONTENTS

<i>Certificate</i>	
<i>Acknowledgement</i>	i
Contents	ii
<i>Abstract</i>	v
<i>Acronyms and Abbreviations</i>	vii
<i>Nomenclatures</i>	ix
<i>List of Figures</i>	xii
<i>List of Tables</i>	xv

1. INTRODUCTION

1.1 Background	1
1.2 Noise Control Techniques	1
1.2.1 Passive Noise Control	1
1.2.2 Active Noise Control	2
1.3 Classification of Noise	6
1.3.1 Broadband Noise	6
1.3.2 Narrowband Noise	6
1.4 Adaptive Active Noise Control	6
1.4.1 Adaptive Algorithms for ANC	7
1.4.2 Multiple-Channel ANC	7
1.4.3 Nonlinear ANC	8
1.4.4 ANC Applications	8
1.5 Motivation	9
1.5.1 Nonlinearity Effect	9
1.5.2 Secondary Path Effect	9
1.5.3 Computational Overload	10
1.5.4 Feedback Effect	10
1.6 Objective and Scope	10
1.7 Organization of the Thesis	11
1.8 Contribution of the Thesis	12
1.9 Summary	13

2. STUDY AND APPLICATIONS OF ACTIVE NOISE CONTROL

2.1 Background	14
2.2 Broadband Feedforward Control	15
2.2.1 FXLMS Algorithm	18
2.2.2 FELMS Algorithm	21
2.2.3 Feedback Effect	22
2.2.4 ANC using IIR Filter	24
2.2.5 Narrowband Feedforward Control	26
2.3 Feedback Control	28
2.4 Multichannel Active Noise Control	28
2.5 Virtual Sensor Technique (VST)	29
2.6 Frequency Domain Approach	32
2.7 Active Noise Control in Headset	33
2.8 ANC in functional Magnetic Resonance Imaging (fMRI)	34
2.9 ANC in Infant Incubators	35
2.10 Study on Nonlinear ANC	36
2.10.1 Adaptive Volterra Filter (AVF) Approach	38
2.10.2 Adaptive Bilinear Filter (ABF) Approach	41
2.10.3 Radial Basis Function Approach (RBF) Approach	44
2.10.4 Fuzzy Logic (FL) Approach	45
2.10.5 Fuzzy-Neural Approach	47
2.10.6 Genetic Algorithm (GA) Approach	49
2.10.7 Particle Swarm Optimization (PSO) Approach	50
2.11 Summary	51

3. NEURAL NETWORK APPROCH TO NONLINEAR ACTIVE NOISE CONTROL

3.1 Background	53
3.2 Multilayer Artificial Neural Network	53
3.3 Proposed Neural Network Technique	56
3.3.1 Linear Secondary Path	59
3.3.2 Nonlinear Secondary Path	61
3.4 Development of Neural Filtered-e LMS Algorithm	64
3.4.1 Linear Secondary Path	65
3.4.2 Nonlinear Secondary Path	65
3.5 Simulation and Results	66
3.6 Summary	84

4. LEGENDRE NEURAL NETWORK FOR NONLINEAR ACTIVE NOISE CONTROL

4.1 Background	85
4.2 Reduced Structure Legendre Neural Network for Nonlinear ANC	87
4.2.1 Legendre Polynomial	87

4.2.2 Legendre Neural Network	89
4.3 LFXLMS Algorithm	90
4.3.1 Nonlinear Secondary Path	92
4.4 LFELMS Algorithm	93
4.5 LFXRLS Algorithm	96
4.6 Fast LFXLMS Algorithm	96
4.7 Simulation and Results	101
4.8 Summary	113

5. FREQUENCY-DOMAIN APPROACH TO MULTICHANNEL ACTIVE NOISE CONTROL

5.1 Background	115
5.2 Filtered-x Least Mean Square (FXLMS) Algorithm	116
5.3 Time Domain Block Filtered-x LMS (BFXLMS) Algorithm	118
5.4 Frequency Domain Block Filtered-x LMS (FBFXLMS) Algorithm	119
5.5 Frequency Domain Block Filtered-x NLMS (FBFXNLMS) Algorithm	120
5.5.1 Reduced Structure FBFXNLMS Algorithm	120
5.6 Frequency Domain Block Filtered-e LMS (FBFELMS) Algorithm	123
5.7 Multichannel FBFXNLMS Algorithm	127
5.7.1 Computational Complexity	127
5.8 Frequency Domain Implementation of Legendre Neural Network for Nonlinear ANC	129
5.9 Frequency Domain Block Legendre FELMS (FBLFELMS) Algorithm	132
5.10 Simulation and Results	138
5.11 Summary	152

6. CONCLUSION AND SCOPE FOR FURTHER RESEARCH

6.1 Conclusion	154
6.2 Scope for Further Research	156

REFERENCE	157
------------------	------------

Bio-data of the Candidate

ABSTRACT

Active noise control has been a field of growing interest over the past few decades. The challenges thrown by active noise control have attracted the notice of the scientific community to engage them in intense level of research. Cancellation of acoustic noise electronically in a simple and efficient way is the vital merit of the active noise control system. A detailed study about existing strategies for active noise control has been undertaken in the present work. This study has given an insight regarding various factors influencing performance of modern active noise control systems. The development of new training algorithms and structures for active noise control are active fields of research which are exploiting the benefits of different signal processing and soft- computing techniques. The nonlinearity contributed by environment and various components of active noise control system greatly affects the ultimate performance of an active noise canceller. This fact motivated to pursue the research work in developing novel architectures and algorithms to address the issues of nonlinear active noise control.

One of the primary focus of the work is the application of artificial neural network to effectively combat the problem of active noise control. This is because artificial neural networks are inherently nonlinear processors and possesses capabilities of universal approximation and thus are well suited to exhibit high performance when used in nonlinear active noise control. The present work contributed significantly in designing efficient nonlinear active noise canceller based on neural network platform. Novel neural filtered-x least mean square and neural filtered-e least mean square algorithms are proposed for nonlinear active noise control taking into consideration the nonlinear secondary path. Employing Legendre neural network led the development of a set new adaptive algorithms such as Legendre filtered-x least mean square, Legendre

filtered-e least mean square, Legendre filtered-x recursive least square and fast Legendre filtered-x least mean square algorithms. The proposed algorithms outperformed the existing standard algorithms for nonlinear active noise control in terms of steady state mean square error with reduced computational complexity. Efficient frequency domain implementation of some the proposed algorithms have been undertaken to exploit its benefits. Exhaustive simulation studies carried out have established the efficacy of the proposed architectures and algorithms.

ACRONYMS AND ABBREVIATIONS

ABF	Adaptive Bilinear Filter
AFC	Adaptive Feedback Cancellation
AGA	Adaptive Genetic Algorithm
ANC	Active Noise Control
ANFIS	Adaptive Neuro Fuzzy Inference System
AVF	Adaptive Volterra Filter
BFXLMS	Block Filtered-x Least Mean Square
CRPSO	Conditional Reinitialized Particle Swarm Optimization
dB	Decibel
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
EFNN	Enhanced Fuzzy Neural Network
EPI	Eco Planner Imaging
FAGA	FIR Adaptive Genetic Algorithm
FBFELMS	Frequency Domain Block Filtered-e Least Mean Square
FBFXLMS	Frequency Domain Block Filtered-x Least Mean Square
FBFXNLMS	Frequency Domain Block Filtered-x Normalized Least Mean square
FBLFXLMS	Frequency Domain Block Legendre Filtered-x Normalized Least Mean Square
FBLFELMS	Frequency Domain Block Legendre Filtered-e Normalized Least Mean Square
FELMS	Filtered-e Least Mean Square
FFELMS	FLANN Filtered-e Least Mean Square
FFT	Fast Fourier Transform
FFXLMS	FLANN Filtered-x Least Mean Square
FIR	Finite Impulse Response
FL	Fuzzy Logic
FLANN	Functional Link Artificial Neural Network
fMRI	functional Magnetic Resonance Imaging
FNN	Fuzzy Neural Network
FPGA	Field Programmable Gate Array
FSLMS	Filtered-s Least Mean Square
FULMS	Filtered-u Least Mean Square
FXLMS	Filtered-x Least Mean Square
FXNLMS	Filtered-x Normalized Least Mean Square

FXRBF	Filtered-x Radial Basis Function
GA	Genetic Algorithm
Hz	Hertz
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response
LFELMS	Legendre Filtered-e Least Mean Square
LFXLMS	Legendre Filtered-x Least Mean Square
LFXRLS	Legendre Filtered-x Recursive Least Square
LMS	Least Mean Square
LNN	Legendre Neural Network
LNL	Linear Nonlinear Linear
LSP	Linear Secondary Path
MIMO	Multiple Input Multiple Output
MLANN	Multilayer Artificial Neural Network
MLP	Multilayer Perceptron
MSE	Mean Square Error
MSE(dB)	Mean Square Error in decibel
MR	Magnetic Resonance
NANC	Nonlinear Active Noise Control
NFELMS	Neural Filtered-e Least Mean Square
NFXLMS	Neural Filtered-x Least Mean Square
NICU	Neonatal Intensive Care Unit
NLMS	Normalized Least Mean Square
NN	Neural Network
NSP	Nonlinear Secondary Path
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RMT	Remote Microphone Technique
RNN	Recurrent Neural Network
SPE	Secondary Path Equalization
S/P	Serial to Parallel
P/S	Parallel to Serial
TDNN	Time Delay Neural Network
VAGA	Volterra Adaptive Genetic Algorithm
VC	Visual Cortex
VFELMS	Volterra filtered-e LMS
VFXLMS	Volterra filtered-x LMS
VST	Virtual Sensor Technique
WI-FXLMS	Wiener Initialized Filtered-x Least Mean Square

NOMENCLATURES

$A(z)$	Primary path transfer function
$\hat{A}_v(z)$	Estimated primary path transfer function between physical and virtual location
$a(n)$	Impulse response of primary path
$\hat{a}_v(n)$	Impulse response of filter having transfer function $\hat{A}_v(z)$
$B(z)$	Secondary path transfer function
$\hat{B}(z)$	Estimated secondary path transfer function
$B_p(z)$	Secondary path transfer function at physical location
$\hat{B}_p(z)$	Estimated Secondary path transfer function at physical location
$\hat{B}_v(z)$	Estimated Secondary path transfer function at virtual location
$\hat{B}_-(z)$	Adjoint Secondary path filter
$b(n)$	Impulse response of secondary path
$b_k(n)$	k^{th} element of secondary path impulse response
$\tilde{\mathbf{b}}(n)$	Impulse response of virtual secondary path
$\mathbf{b}_-(n)$	Impulse response of adjoint secondary path
$\tilde{b}_k(n)$	k^{th} element of virtual secondary path impulse response
$\tilde{\mathbf{b}}_-(n)$	Impulse response of adjoint virtual secondary path filter
$W(z)$	Transfer function of adaptive filter
$x(n)$	Reference signal
$x'(n)$	Filtered Reference signal
$d(n)$	Undesired signal
$\hat{d}(n)$	Estimated undesired signal
$\hat{d}_p(n)$	Estimate of primary disturbance at physical location
$\hat{d}_v(n)$	Estimate of primary disturbance at virtual location
$d_p(n)$	Primary disturbance at physical location
$y(n)$	Adaptive filter output
$y_p(n)$	Antinoise at physical location
$\hat{y}_p(n)$	Estimated antinoise at physical location

$\hat{y}_v(n)$	Estimated antinoise at virtual location
$e(n)$	Error signal
$e_p(n)$	Error signal from physical microphone
$\hat{e}_v(n)$	Estimated error at virtual location
n	Time index
i	An arbitrary variable
j	An arbitrary variable
l	An arbitrary variable
p	An arbitrary variable
q	An arbitrary variable
L	Number of reference microphones
P	Number of secondary loudspeakers
Q	Number of error microphones
$\mathbf{w}(n)$	Coefficient vector of adaptive filter
$w_{i,j}(n)$	Filter weights
$\mathbf{x}(n)$	Reference signal vector
$\mathbf{x}'(n)$	Filtered reference signal vector
$\mathbf{X}(n)$	FFT of reference signal vector
$\mathbf{X}'(n)$	FFT of filtered reference vector
$\mathbf{Y}(n)$	FFT of adaptive filter output vector
$\mathbf{E}(n)$	FFT of flipped error signal vector
$\mathbf{W}(n)$	FFT of coefficient vector of adaptive filter
$\xi(n)$	Cost function
N	Order of adaptive filter
μ	Step size
μ_{\max}	Maximum allowable step size for FXLMS algorithm
$\nabla \xi(n)$	Instantaneous estimate of gradient on MSE surface
$P_{x'}$	Power of the filtered reference signal
$E[.]$	Expectation operator
Δ	No of samples corresponding to overall delay in the secondary path
$e'(n)$	Filtered error signal
$f(n)$	Impulse response of the feedback path
$\hat{f}(n)$	Estimated Impulse response of the feedback path
$F(n)$	Transfer function of the feedback path
$\hat{F}(n)$	Estimated transfer function of the feedback path
z^{-1}	Unit delay
$W_f(z)$	Forward path transfer function of IIR filter
$W_b(z)$	Feedback path transfer function of IIR filter

$\mathbf{w}_f(n)$	Weight vector of $W_f(z)$
$\mathbf{w}_b(n)$	Weight vector of $W_b(z)$
L_{w_f}	Order of $W_f(z)$
L_{w_b}	Order of $W_b(z)$
$\mathbf{u}(n)$	Reference vector
$\mathbf{u}'(n)$	Filtered reference vector
$y(n)$	Output signal of adaptive filter
$y'(n)$	Filtered output signal of adaptive filter
$\mathbf{y}(n)$	Output signal vector of adaptive filter
$\mathbf{y}'(n)$	Filtered output signal vector of adaptive filter
I	Number of neurons in input layer
J	Number of neurons in hidden layer
$v_j(n)$	Synaptic weights of output layer
$w_{ij}(n)$	Synaptic weights of hidden layer
F	Activation function
$u_j(n)$	Hidden layer node output
$c_j(n)$	Net internal activity level of neural node
M	Order of secondary path filter
$L_p(x)$	Legendre Polynomial
$\mathbf{s}_p(n)$	Expanded reference vector
$\mathbf{s}'_p(n)$	Filtered expanded reference vector
$\mathbf{k}_i(n)$	Kalman gain vector
λ	Forgetting factor
$\mathbf{Q}_i(n)$	Inverse of the autocorrelation matrix
$\boldsymbol{\alpha}_{i,j}$	Expanded input signal vector
$\boldsymbol{\psi}_i(n)$	Expanded input signal
$\boldsymbol{\beta}_{i,j}$	Expanded input signal vector
$\boldsymbol{\gamma}_{i,j}$	Expanded input signal vector
T	Vector transpose
A_i	Number of additions
M_i	Number of multiplications
\mathbf{I}_N	$N \times N$ identity matrix
\mathbf{O}_N	$N \times N$ matrix with all zero elements
F_{2N}	2N-point FFT operator
F_{2N}^{-1}	2N-point IFFT operator

LIST OF FIGURES

1.1 Lueg's active noise control patent (U.S. patent No. 2043416, June 1936)	3
1.2 Principle of active noise control	4
1.3 Diagram of the basic arrangement of an ANC in a duct	5
2.1 Schematic diagram of a broadband feedforward ANC	15
2.2 Block diagram of a single channel feedforward ANC	16
2.3 Impulse response of primary path	17
2.4 Impulse response of secondary path	17
2.5 Block diagram of ANC using adjoint LMS algorithm	22
2.6 Schematic diagram of ANC with feedback neutralization filter	23
2.7 Block diagram of ANC with feedback neutralization filter	24
2.8 Block diagram of ANC (with feedback) using IIR adaptive filter	25
2.9 Schematic diagram of a narrowband feedforward ANC	27
2.10 Schematic diagram of a feedback ANC	28
2.11 Schematic diagram of a multi-channel ANC for an enclosure	29
2.12 Schematic diagram of ANC using virtual sensor technique	30
2.13 Block diagram of ANC using virtual sensor technique	31
2.14 Block diagram of ANC using frequency-domain FXLMS algorithm	32
2.15 Schematic diagram of ANC used for headphone or headset	34
2.16 Block diagram of ANC using filtered-x second order Volterra adaptive algorithm	38
2.17 Block diagram of ANC using adaptive output-error bilinear filter	42
2.18 Block diagram of ANC using RBF	44
2.19 Structure of the two RBF networks for ANC	44
2.20 Structure of the two recurrent RBF networks for ANC	45
2.21 Block diagram of ANC using fuzzy logic	46
2.22 Structure of the neuro-fuzzy controller used for ANC	47
2.23 Block diagram of ANC using ANFIS	47
2.24 Block diagram of ANC using fuzzy-neural network	48
2.25 Structure of the fuzzy-neural network for ANC	48
2.26 Block diagram of the genetic ANC system	49
2.27 Block diagram of PSO-based training of an ANC system	51

3.1 Block diagram of ANC using two neural networks	55
3.2 Block diagram of ANC using neural network	56
3.3 Neural network controller	57
3.4 Block diagram of ANC using FELMS algorithm	64
3.5 MSE(dB) plot for NFXLMS and VFXLMS algorithm (activation function I)	68
3.6 MSE(dB) plot for NFXLMS and VFXLMS algorithm (activation function II)	69
3.7 MSE(dB) plot for NFXLMS and VFXLMS algorithm (activation function III)	69
3.8 MSE(dB) plot for NFELMS and VFELMS algorithm (activation function I)	70
3.9 MSE(dB) plot for NFELMS and VFELMS algorithm (activation function II)	70
3.10 MSE(dB) plot for NFELMS and VFELMS algorithm (activation function III)	71
3.11 MSE(dB) plot for NFXLMS and VFXLMS algorithm (activation function I)	73
3.12 MSE(dB) plot for NFXLMS and VFXLMS algorithm (activation function II)	73
3.13 MSE(dB) plot for NFXLMS and VFXLMS algorithm (activation function III)	74
3.14 MSE(dB) plot for NFELMS and VFELMS algorithm (activation function I)	74
3.15 MSE(dB) plot for NFELMS and VFELMS algorithm (activation function II)	75
3.16 MSE(dB) plot for NFELMS and VFELMS algorithm (activation function III)	75
3.17 Factory Floor Noise	77
3.18 Buccaneer Jet Cockpit Noise	77
3.19 MSE(dB) plot for NFXLMS and VFXLMS algorithms(Factory Floor Noise)	79
3.20 MSE(dB) plot for NFXLMS and VFXLMS algorithms(Jet Cockpit Noise)	79
3.21 MSE(dB) plot for NFELMS and VFELMS algorithms(Factory Floor Noise)	80
3.22 MSE(dB) plot for NFELMS and VFELMS algorithms(Jet Cockpit Noise)	80
3.23 MSE(dB) plot for NFXLMS and VFXLMS algorithms(Factory Floor Noise)	82
3.24 MSE(dB) plot for NFXLMS and VFXLMS algorithms(Jet Cockpit Noise)	82
3.25 MSE(dB) plot for NFELMS and VFELMS algorithms(Factory Floor Noise)	83
3.26 MSE(dB) plot for NFELMS and VFELMS algorithms(Jet Cockpit Noise)	83
4.1 Legendre polynomial expansion	88
4.2 Legendre neural network	89
4.3 Block diagram of ANC using reduced structure Legendre neural network	90
4.4 Secondary path filter	94
4.5 Adjoint Secondary path filter	94
4.6 Virtual Secondary path filter	95
4.7 Adjoint virtual Secondary path filter	95
4.8 MSE(dB) plot for LFXLMS, FFXLMS, LFELMS and LFXRLS algorithm	104
4.9 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm when primary path is changed at 3000 th iteration	104
4.10 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm when secondary path is changed at 3000 th iteration	105
4.11 MSE(dB) plot for LFXLMs, FFXLMS and LFELMS algorithm when	

both primary path and secondary path are changed at 3000 th iteration	105
4.12 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm	106
4.13 Block diagram of LNL nonlinear secondary path model	106
4.14 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm	107
4.15 MSE(dB) plot for LFXLMS and fast LFXLMS algorithm	108
4.16 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm(Jet Cockpit Noise)	110
4.17 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm(White Noise)	110
4.18 MSE(dB) plot for LFXLMS and FFXLMS algorithm when primary path is changed at 3000 th iteration(Jet Cockpit Noise)	111
4.19 MSE(dB) plot for LFXLMS and FFXLMS algorithm when primary path is changed at 3000 th iteration(White Noise)	111
4.20 MSE(dB) plot for LFXLMS and FFXLMS algorithm when secondary path is changed at 3000 th iteration(Jet Cockpit Noise)	112
4.21 MSE(dB) plot for LFXLMS and FFXLMS algorithm when secondary path is changed at 3000 th iteration(White Noise)	112
4.22 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm(Jet Cockpit Noise)	113
4.23 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm(White Noise)	113
5.1 Block diagram of basic active noise control system	116
5.2 Block diagram of ANC using FBFXNLMS algorithm	121
5.3 Block diagram of ANC using reduced structure FBFXNLMS algorithm	122
5.4 Block diagram of ANC using FBFELMS algorithm	124
5.5 Comparison computational complexity (a) multiplication (b) addition	129
5.6 Legendre neural network for nonlinear ANC	130
5.7 Frequency domain implementation of Legendre neural network for ANC	133
5.8 details of T block of fig. 5.7 (FBLFXLMS algorithm)	134
5.9 Details of T block of fig. 5.7 (FBLFELMS algorithm)	135
5.10 MSE(dB) plot of FBFXNLMS algorithm for multichannel ANC	139
5.11 MSE(dB) plot of FXNLMS algorithm for multichannel ANC	139
5.12 MSE(dB) plot of FBFXNLMS algorithm for multichannel ANC	140
5.13 MSE(dB) plot of FXNLMS algorithm for multichannel ANC	140
5.14 MSE(dB) plot of FBLFELMS algorithm at error microphone-1	143
5.15 MSE(dB) plot of FBLFELMS algorithm at error microphone-2	143
5.16 MSE(dB) plot of FBLFELMS algorithm at error microphone-3	144
5.17 MSE(dB) plot of FBLFELMS algorithm at error microphone-4	144
5.18 MSE(dB) plot of FBLFELMS algorithm of all the error microphones	145
5.19 MSE(dB) plot of LFELMS algorithm of all the error microphones	145
5.20 MSE(dB) plot of FXNLMS algorithm for white noise	147
5.21 MSE(dB) plot of FBFXNLMS algorithm for white noise	147
5.22 MSE(dB) plot of FXNLMS algorithm for factory floor noise	148

5.23 MSE(dB) plot of FBFXNLMS algorithm for factory floor noise	148
5.24 MSE(dB) plot of FXNLMS algorithm for Buccaneer jet cockpit noise	149
5.25 MSE(dB) plot of FBFXNLMS algorithm for Buccaneer jet cockpit noise	149
5.26 MSE(dB) plot of FBLFELMS algorithm for Buccaneer cockpit noise at error microphone-1	150
5.27 MSE(dB) plot of FBLFELMS algorithm for Buccaneer cockpit noise at error microphone-2	150
5.28 MSE(dB) plot of FBLFELMS algorithm for Buccaneer cockpit noise at error microphone-3	151
5.29 MSE(dB) plot of FBLFELMS algorithm for Buccaneer cockpit noise at error microphone-4	151
5.30 MSE(dB) plot of FBLFELMS algorithm of all the error microphones (combined)	152

LIST OF TABLES

4.1 Computational complexity comparison	101
5.1 FBFXNLMS algorithm	125
5.2 FBFELMS algorithm	126
5.3 Computational complexity per samples	128
5.4 FBLFXLMS algorithm	136
5.5 FBLFELMS algorithm	137

Chapter 1

Introduction

1.1 Background

Industrialization and ever increasing use of technology in our daily life has led to exponential increase of acoustical noise pollution. Noise is an irritant and causes mental strain. It has an adverse psychological effect on living beings and creates an unhealthy working environment. Continuous exposure to noise has a detrimental effect leading to temporary or permanent loss of hearing and mood swing. Acoustic noise problem in the environment is gaining attention due to the tremendous growth of technology that has led to noisy engines, heavy machinery, pumps, home appliances and a myriad other noise sources. Exposure to high levels of sound proves damaging to humans from both physical and psychological aspect. The problem of controlling the noise level in the environment has been the focus of a tremendous amount of research over the years. Legislations have been enforced on industries and manufacturers to keep the maximum noise level of their products under specified limits. However, as long as the quest for larger machinery using light material continues, the noise pollution level will be on the rise. Due to these reasons, noise control has gained considerable importance in the recent years. Human desire for a high-tech but comfortable and noise free living has fueled the development of acoustic noise control techniques.

Acoustic noise control techniques can be broadly classified into two categories: passive control technique and active control technique. The combination of passive control technique with active control technique can yield better overall performance than for either one alone. In many respects, active noise control (ANC) is a complementary technology to passive silencing [1].

1.2 Noise Control Techniques

1.2.1 Passive Noise Control

Passive noise control techniques employ sound absorbing material, enclosures, barriers and silencers to attenuate the undesired noise. These passive techniques are

effective for high attenuation over a broad frequency range; but are unable to absorb low frequency noise. For low frequency noise the passive techniques become relatively larger and heavier thus considerably increasing the cost. This often makes the passive approach to reduce low frequency noise very impractical. But the fact is people are more uncomfortable with low frequency noise rather than high frequency noise because low frequency noise is not only annoying but produces fatigue, irritation and loss of concentration, therefore affecting productivity. If low frequency noise is mixed with speech, it reduces speech intelligibility [1], [2]. In a noise generating system the amplitude of the low frequency noise is mostly higher than other frequencies. Hence there is a growing demand for reducing low frequency noise.

1.2.2 Active Noise Control

The design of an active noise controller using a microphone and an electronically driven loudspeaker to generate a cancelling sound was first proposed and patented using a purely analog electronic approach in 1930 in France by Coanda [4] and in US by Lueg in 1936 [5]. The patent outlined the basic idea of ANC, shown in fig.1.1, but at that time it could not be applied to practice because of a number of factors. Advanced and accurate electronic instruments (microphone, loudspeakers, digital personal computers) were not available, digital signal processing and the concept of adaptive systems were not started then and there was no proper knowledge on noise and various sources of noise. During the latter half of the 20th century, emergence of digital signal processing made ANC a viable technique for practical noise reduction. Latter on advances in adaptive systems and adaptive signal processing which facilitate a time varying system with the ability to adapt to changing environment, not only revolutionized ANC but also further opened up its field of applications. Development of high speed special purpose digital signal processors helped realize practical ANC. More recently fusion of the soft computing approaches like artificial neural network, fuzzy logic and hybrid techniques (neuro-fuzzy techniques) with existing ANC techniques have made ANC powerful than ever before. ANC is currently being researched for use to control noise from jet engines, helicopter, motor vehicle engines, ventilation systems, generators, transformers, industrial machinery, traffic, MRI units, torpedoes, headphones and even amplified music.

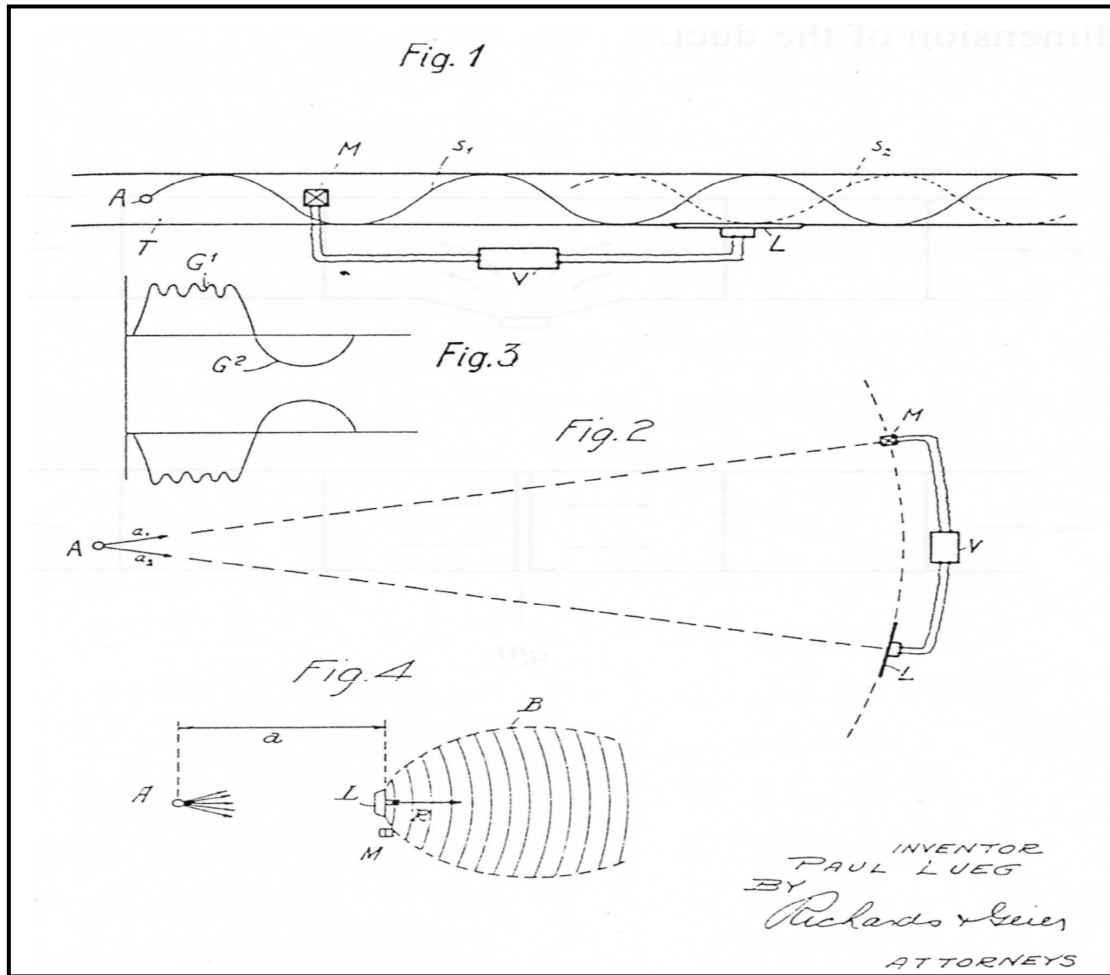


Fig 1.1 Lueg's Active Noise Control Patent (U.S. Patent No. 2043416, June 1936) [5].

The ANC is an electroacoustic device that is based on the principle of destructive interference where the unwanted sound is cancelled by generating an antinoise (antinoise) of equal amplitude and opposite phase. The original unwanted noise and the antinoise superimpose acoustically, resulting in the cancellation of both sounds. For example, fig. 1.2 shows the waveforms of a typical unwanted noise (called the reference noise), the cancelling noise (called the antinoise), and the residual noise that results when they superimpose. For cancellation of the reference noise the amplitude of antinoise must be same as that of reference noise but phase should be opposite (180° out of phase). So the effectiveness of cancellation of the reference noise depends on the accuracy of the amplitude and phase of the generated antinoise.

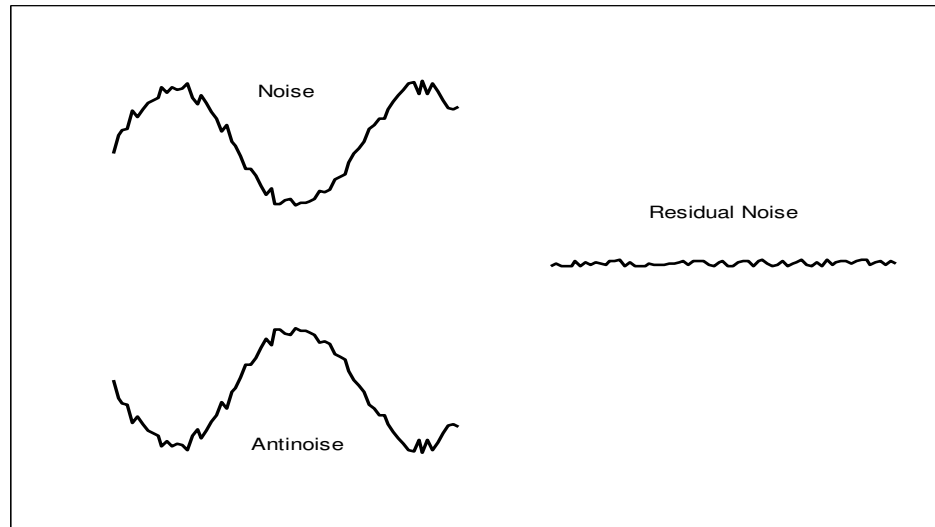


Fig 1.2 Basic Principle of Active Noise Control.

As discussed earlier passive noise control approach is effective for controlling noise over the entire frequency range except at low frequency (below 600 Hz). Hence the demand is to develop ANC which should be able to control low frequency noise. ANC take advantage of this situation as lower noise frequency allow lower sampling rate (as low as 1200 Hz) resulting in comparatively higher sample period (approximately 0.833millisecond for a sampling frequency of 1200Hz). An ANC has to finish all the computation to generate antinoise within a single sample period and should be ready to accommodate the next sample of reference signal as soon as it arrives (reference noise signal samples are available at the rate of one sample every sample period). A larger signal sample period not only allow the ANC to finish it computation in time but also facilitates employment of more complicated and computationally intensive adaptive algorithms for achieving better noise cancellation performance.

So active attenuation is an attractive means to achieve large amounts of noise reduction electronically, particularly at low frequencies. Another advantage of low frequencies noise is, it allow plane wave propagation making the job of an ANC easier as the sound field is not complicated. In essence, active noise control shows real advantages to control low frequency noise. So ANC has received considerable research interest and has shown significant potential to control low frequency noise. The creation and superpo-

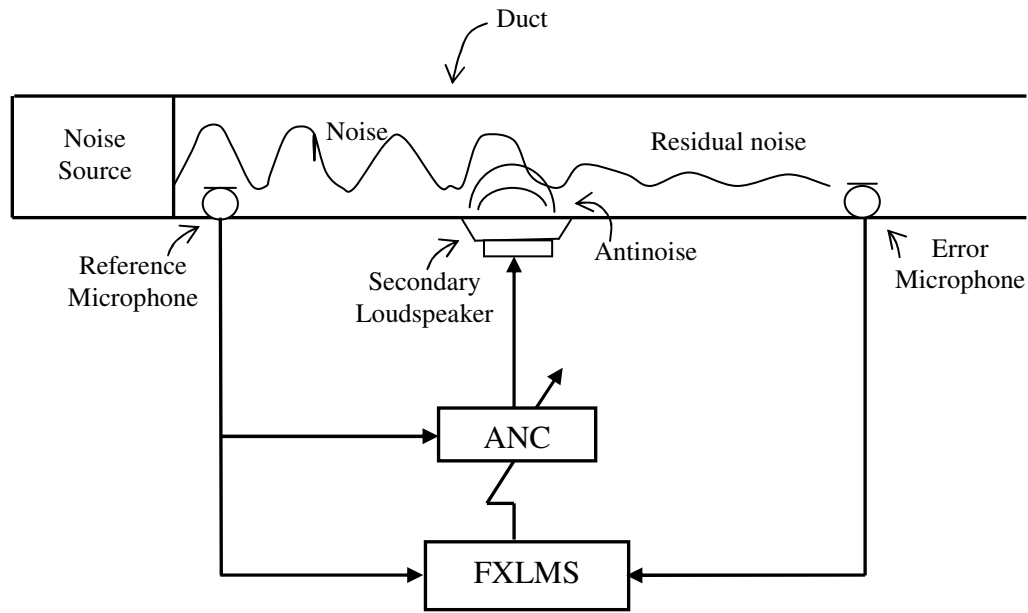


Fig.1.3 Diagram of the basic arrangement for an ANC in a duct.

sition of the antinoise for controlling three dimensional reference noise is very complex, since this involves reconstruction of the whole acoustic event. The discussion in this thesis restricts itself to one-dimensional ANC in long ducts. One basic arrangement to cancel noise in a duct is shown in the fig. 1.3.

Working with low frequency noise in ducts has got the following advantages: firstly, the sound will travel as plane waves upto a certain frequency called cutoff frequency. The noise of higher frequencies will decay within a short distance from the source. So the mixing of the reference noise and antinoise is easier. Secondly, the low frequency noise has a longer wavelength, so that the phase angle changes slowly with time. This makes the fine control of phase of secondary wave easier. Hence, a stable interference pattern is possible that results in larger noise reduction. Lastly, the sound wave travels at much slower speed than the electrical signals, so that a large operation time for generating the antinoise is available, if secondary source is suitably located. Acoustic noise can be broadly classified into two types: broadband noise and narrowband noise.

1.3 Classification of Noise

1.3.1 Broadband Noise

Broadband noise is caused by turbulence and is random in nature. Turbulent noise distributes its energy evenly across the frequency bands. The examples are the low-frequency sounds of jet planes and the impulse noise of an explosion.

1.3.2 Narrowband Noise

In narrowband noise most of its energy is concentrated at specific frequencies. This type of noise is related to rotating or repetitive machines, so it is periodic or nearly periodic in nature. Examples of narrowband noise include the noise of internal combustion engines in transportation, compressors as auxiliary power sources and in refrigerators, and vacuum pumps used to transfer bulk materials in many industries. The transformer noise which is the hum noise due to the magnetostriction consists of higher harmonics of the power-line frequency. In another way the noise can be classified into two types: linear noise and nonlinear noise. The broadband noise is mostly linear. But there are situations where noise coming from a dynamic system may be nonlinear and deterministic. Such nonlinear but deterministic noise is referred to as chaotic noise. Some examples of chaotic noise are Logistic chaotic noise, Lorenz chaotic noise and Duffing chaotic noise [11]. One practical example of chaotic noise is the fan noise which often shows chaotic behavior.

1.4 Adaptive Active Noise Control

The acoustic noise source and the environment are time varying, the frequency content, amplitude, phase, and velocity of the undesired noise are nonstationary (time varying). So an active noise control system must be adaptive in order to cope with these changing characteristics. This is the reason why modern active noise control systems depend heavily on digital signal processing because in the field of digital signal processing, there are classes of systems called adaptive systems which have the capability to vary their coefficients in order to cope with changing environment. Adaptive systems can be implemented as transversal—finite impulse response (FIR), recursive—infinite impulse response (IIR), lattice filters, transform-domain filters. Correspondingly ANC can also be implemented using FIR filter, IIR filter, lattice filters or transform-

domain filters. These yield varieties of ANC different from each other in many aspects. Performance of each of them is well researched and documented in literature [1].

The coefficients of an adaptive filter are tuned to minimize a predefined cost function which is generally a function of error signal. The process of adaptation is automated by DSP adaptation algorithms. Depending on the problem, a large number of adaptation algorithms have been developed to adapt the system quickly and efficiently. The pros and cons of available adaptive algorithms have been analyzed mathematically, through exhaustive computer simulation and also real time implementations. A large number of researchers are still involved in developing new adaptation algorithms by exploring advanced digital signal processing techniques and soft computing techniques in order to optimize the overall performance of the ANC.

1.4.1 Adaptive Algorithms for ANC

The ANC system can be implemented using different adaptive learning algorithms. The most common algorithm applied to adaptive filters is the least mean-squared (LMS) algorithm [1]-[3],[7]. The reference noise signal and error signal are used as input to an adaptive algorithm, which adjusts the adaptive filter coefficients to model (estimate) the acoustic-channel effects. For ANC, taking into account presence of secondary path, LMS algorithm is suitably modified to develop an efficient but simple algorithm known as filtered-x LMS (FXLMS) algorithm which was derived by Widrow [1]. With advances in digital signal processing, FXLMS algorithm is further modified by many researchers to improve the ANC overall performance. All the developed algorithms have their relative merits and demerits in terms of speed of convergence, residual noise, computational complexity, stability and robustness.

1.4.2 Multiple-Channel ANC

In applications where noise field is complex or the required zone of silence is quite large, use of single reference microphone, single secondary loudspeaker or an error microphone is not sufficient to reduce noise to a desirable limit. In this scenario, several numbers of secondary loudspeakers, error microphones and reference microphones are employed. The most important aspect is proper positioning of sensors to have overall global noise suppression and also to optimize the service of each sensor.

1.4.3 Nonlinear ANC

FXLMS algorithm and all its variants are useful for linear ANC's i.e. when the primary path, secondary path and the reference noise are linear. But practically there are many sources of nonlinearity which exist in an ANC. So most of the ANC's are found to be nonlinear in nature hence the present arrangement with linear FXLMS algorithm or all its variants exhibit performance degradation. There is ample scope for performance improvement by employing various nonlinear structures and algorithms. Volterra filter is a nonlinear adaptive filter best suited for this type of problem. The real advantage of using Volterra adaptive filter lies with the fact that it can be trained by linear type adaptive algorithms. So FXLMS algorithm and most of its variants can be extended to train ANC designed using Volterra adaptive filter.

Different neural networks such as multi layer perceptron (MLP) with derivative based back propagation training algorithm, radial basis function networks, fuzzy logic and neuro-fuzzy have also been used for nonlinear ANC. Research is in nascent stage for using derivative free evolutionary techniques such as genetic algorithm and particle swarm optimization for nonlinear ANC.

ANC systems may soon be available for many noisy environments. As industry adopts mechatronic design techniques, acoustic considerations can be made from the outset of the design process. ANC can be expected to be an integral part of vehicle and industrial design.

1.4.4 ANC Applications

ANC applications can be broadly classified under the following categories

- **Duct noise:** This is the major application area of ANC because of widespread industrial applications like heating, ventilating and air conditioning systems. In air conditioning systems noise is controlled using ANC in the ducts, which is the transmission path of noise.
- **Personal hearing protection:** The headphone falls into this classification, where the loudspeaker generates not only the desired sound but also an antinoise which cancel the low frequency ambient noise. Headphones also have ear shells which attenuate high frequency noise.

- Enclosure Noise: Suppression of noise inside an enclosure like the noise inside the passenger cabin of a propeller-driven aircraft is a typical example of enclosure noise ANC.
- Open field noise: Transformer noise cancellation or creating a zone of silence near the ears of a person sitting in a chair by providing two loudspeakers to generate antinoise near the headrest.
- Virtual Microphone: Placing a microphone at the desired place of noise reduction is not always practical like noise in a duct carrying industrial fluid. Virtual acoustic sensor techniques have been developed to overcome this problem. Here virtual acoustic sensors create a zone of silence at the desired location which may be remote from the physical sensor position.

1.5 Motivation

It has been reported in the literatures that various problems are encountered while practical implementation of the ANC systems are carried out. Some of these are:

1.5.1 Nonlinearity Effect

In almost all applications of ANC primary path and secondary path exhibit nonlinear characteristics. In some applications reference noise is also produced by a nonlinear noise process. This is called nonlinear active noise control (NANC). Nonlinearity associated with noise process and the paths create problems in linear adaptation of the ANC. In the recent literatures, several methods have been proposed for NANC. With advances in digital signal processing and soft computing techniques much remain to be done to improve the overall performance of NANC.

1.5.2 Secondary Path effect

In order to enable the adaptive filter to learn properly to a desired solution, it is necessary to compensate for the transfer function of the secondary path, $B(z)$, from the secondary source to the error sensor. Presence of secondary path following the adaptive filter prevents straight forward applications of the adaptive algorithms. This requires careful modification of the algorithms to be successfully applied to ANC. This leads to the modification of least mean square (LMS) algorithm to develop filtered-x least mean

squares (FXLMS) algorithm by Widrow [1]. Development of new algorithms which take care of secondary path is still an open area of research.

1.5.3 Computational Overload

Filtered-x least mean square (FXLMS) algorithm based ANC needs hundreds of taps to realize appreciable noise cancellation. As the number of taps increase the computational complexity of the FXLMS algorithm increases significantly. Also in FXLMS algorithm the reference signal is required to be filtered through the secondary path estimate. In practical ANC system, the secondary path estimate also has hundreds of taps. Hence, involvement of large computational complexity is a major problem with regards to implementation. A number of research works have been undertaken with an aim to reduce the computational complexity and make the ANC computation fast.

1.5.4 Feedback Effect

The acoustic feedback from the loudspeaker to the reference microphone, which causes degradation in the performance of the ANC system, is known as the feedback effect. Many investigators have proposed different structures to circumvent this problem. However, their models are practically not implementable or do not provide perfect cancellation of feedback noise. The online adaptive feedback cancellation (AFC) based ANC system also shows poor performance if the reference signal is narrowband or periodic.

From the literature survey on the topic, it has been concluded that attempts have been made to resolve the associated problems of ANC. However, further scope still exists to improve on the performance by devising efficient techniques to tackle these problems. Hence, these observations provide the motivation to undertake research work on these problems of ANC. The objective of the thesis is to develop novel algorithms using soft computing and DSP techniques to efficiently mitigate various issues of NANC.

1.6 Objective and Scope

Efficient ANC for real time implementation has been a challenging task as evident from the literature survey and has opened up new avenues to develop high performance based ANC for nonlinear environment. The present work primarily focuses in addressing the following issues.

- To study the performance of available nonlinear ANC.

- To develop new ANC meant for nonlinear scenario employing soft computing approaches like artificial neural network.
- Reduction in computational complexity requirement of the ANC without compromising with noise cancellation performance.
- To explore frequency domain implementation of multichannel ANC and nonlinear ANC techniques.

1.7 Organization of the thesis

The work in this thesis is organized as follows:

Chapter-I: This chapter presents an introduction where various problems linked with ANC implementation are discussed. A brief literature survey, motivations for doing research, objective and scope of the thesis and organization of the thesis are also presented.

Chapter-II: This chapter deals with an exhaustive study of existing linear and nonlinear ANC schemes. Various factors affecting the noise cancellation performance are discussed. The role played by ANC and the apparent benefits of employing an ANC in these applications are analyzed extensively. The problems encountered in the real time applications of ANC are also highlighted.

Chapter-III: The multilayer perceptron (MLP) is employed as the controller and new synaptic weight update algorithm is developed. This adaptive algorithm is found to be a generalized version of FXLMS algorithm. In order to reduce complexity, FELMS algorithm is also explored by using adjoint secondary path.

Chapter-IV: A novel reduced structure Legendre neural network for nonlinear ANC is proposed for active mitigation of nonlinear noise processes. Low complexity Legendre filtered-x LMS (LFXLMS) algorithm, Legendre filtered-e LMS (LFELMS) algorithm, Legendre filtered-x RLS (LFXRLS) algorithm are developed.

Chapter-V: Frequency domain block algorithms are developed, both for filtered-x and filtered-e paradigm, basically to reduce computational complexity. The proposed algorithms are extended to incorporate multichannel ANC systems. The Legendre neural network developed in the previous chapter is also implemented in frequency domain.

Chapter-VI: The overall conclusion of the thesis and the scope for further research are outlined in this chapter.

1.8 Contribution of the Thesis

This thesis proposes solutions to a number of burning issues of ANC system and suggests some novel methods using DSP and soft computing techniques to mitigate them. The problems dealt in this thesis are secondary path effect, nonlinearity effect, computational overload and multichannel effect. DSP techniques used in this thesis are adaptive filtering and discrete transforms like FFT. The soft computing tools used are Volterra filters, functional link artificial neural network and Legendre neural network. These are suitably applied to develop efficient ANC systems.

Development of new nonlinear ANC structure and pertinent algorithms is the principal thrust of this study. The multilayer perceptron is used as the controller and associated adaptive algorithms are developed. The new neural based adaptive algorithm is found to be an extended version of FXLMS algorithm. The specific advantage of the developed ANC relies on the fact that it can incorporate nonlinear secondary path without using a second neural network for nonlinear secondary path modeling. This is possible by deriving a time varying virtual secondary path. In an effort to reduce computational complexity FELMS algorithm is also explored by using adjoint secondary path concept. To validate the proposed controller and the algorithms, exhaustive simulation study is carried out. Performances of the proposed algorithms are compared with Volterra based algorithms. Neural based algorithms are found to be clearly outperforming the Volterra based algorithms.

A novel Legendre neural network for nonlinear active noise control is also proposed. Legendre polynomial is used for functional expansion of the reference input of the controller. Filter bank implementation of the controller is carried out and the equations relative to weight adaptation are derived. Various block oriented models such as Linear-Nonlinear-Linear model are successfully used for nonlinear secondary path modeling in case of the developed Legendre neural network for NANC. Recursive least square (RLS) algorithm is also explored to develop Legendre filtered-x RLS (LFXRLS) algorithm. A fast algorithm is also developed which reduces computational complexity by updating the weight vector once in two iterations without sacrificing the system performance.

Frequency domain block algorithms are developed basically to reduce computational complexity. The proposed algorithms are extended to incorporate multichannel ANC systems. The nonlinear ANC using Legendre neural network developed in the present work is also implemented in frequency domain to reduce computational complexity and is found to be an efficient candidate for nonlinear ANC systems.

1.9 Summary

This thesis proposes solutions to a number of critical issues of ANC system and suggests some novel methods using DSP and soft computing techniques to alleviate them. The problems generally considered in this thesis are nonlinearity effect, computational overload and adaptive filtering techniques employing both time domain and frequency domain approaches to study the problems associated with ANC. Further efficient ANC structure designs based on the platforms of neural network, Volterra filters, Legendre neural network and functional link artificial neural network have been developed to achieve better performances.

Chapter 2

Study and Applications of Active Noise Control

2.1 Background

Active noise control is an attractive technique for mitigation of undesirable noise, particularly low frequency noise. ANC should be used in combination with passive noise control to get an overall noise reduction across the entire audible frequency range. Researchers have shown interest continuously for the last few decades to develop efficient ANC. The development was greatly influenced by advances in digital signal processing and adaptive filtering. More recently domain of ANC benefitted hugely by applying soft computing techniques. The present research work is particularly focused on designing better ANC controller employing new architecture/structure and/or developing novel adaptive algorithms. The effort is concentrated in studying the factors which greatly influence the real time application of ANC. Some of these important factors are the steady-state noise reduction capability, rapid convergence and reducing computational complexity to reduce response time.

ANC has been in use and currently being researched to be used in diverse field of engineering and technology. Some of the application areas are, to name a few, heating, ventilating and air conditioning, headphone, earplug (headset), hearing protection, hearing aid, infant incubator, transformer noise reduction, functional Magnetic Resonance Imaging (fMRI), vehicle interior noise reduction, noise reduction in airplane and locomotives, active voice control, noise reduction in industry and mines and military etc.

In this chapter an in-depth study of existing ANC techniques is carried out. The study can be subdivided into two categories

- i) Study on linear ANC
- ii) Study on nonlinear ANC.

Effort has also been made in this chapter to study some of the important applications of ANC till date.

2.2 Broadband Feedforward Control

Structurally active noise control can be implemented in two different ways such as feedforward control and feedback control. Depending on the characteristic of noise to be cancelled feedforward control is further classified as broadband feedforward control system and narrowband feedforward control system. The schematic diagram of a broadband feedforward noise cancellation system in a long duct is shown in fig.2.1.

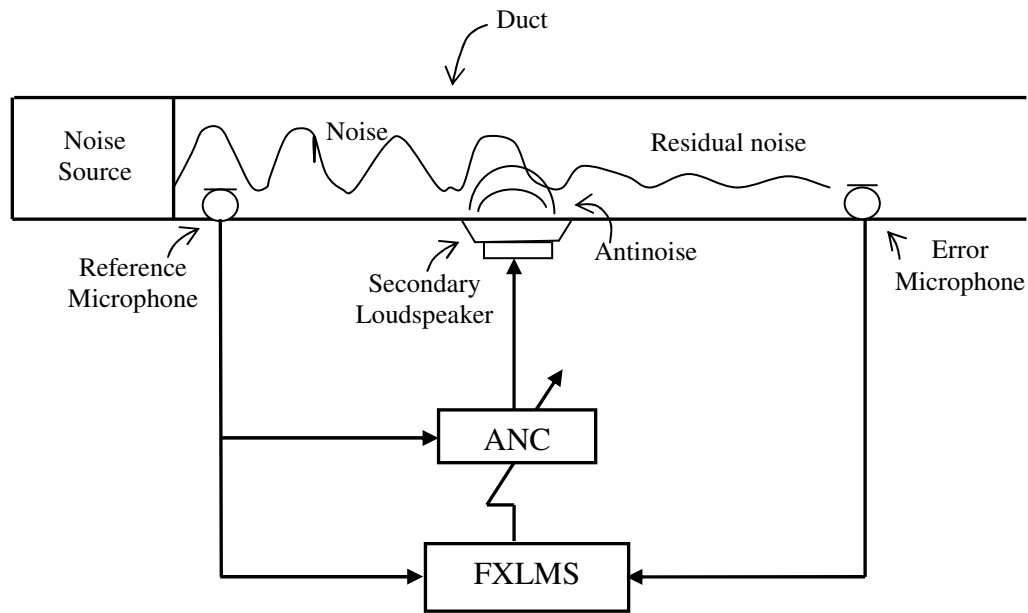


Fig 2.1 Schematic diagram of a broadband feedforward ANC system.

Broadband feedforward ANC system use acoustic sensor (reference microphone) to pickup a coherent reference noise and generates the necessary antinoise (by secondary loudspeaker) before it propagates past the secondary loudspeaker. Broadband noise cancellation requires knowledge of the noise source (the reference noise) in order to generate the antinoise signal. Hence the reference noise is received by a reference microphone and is fed as an input to the noise canceller. After superposition of noise and antinoise most of the noise cancel and a small amount of residual noise may remain which is called as error. This error signal is observed by a microphone called error

microphone. By the time noise canceller generate the antinoise via secondary loudspeaker, the reference noise will travel a distance and reach at the position of secondary loudspeaker. Care should be taken so that there is time alignment between the reference noise reaching the secondary loudspeaker and the antinoise. Any mismatch in time alignment will result in causality problem and performance of ANC will be poor.

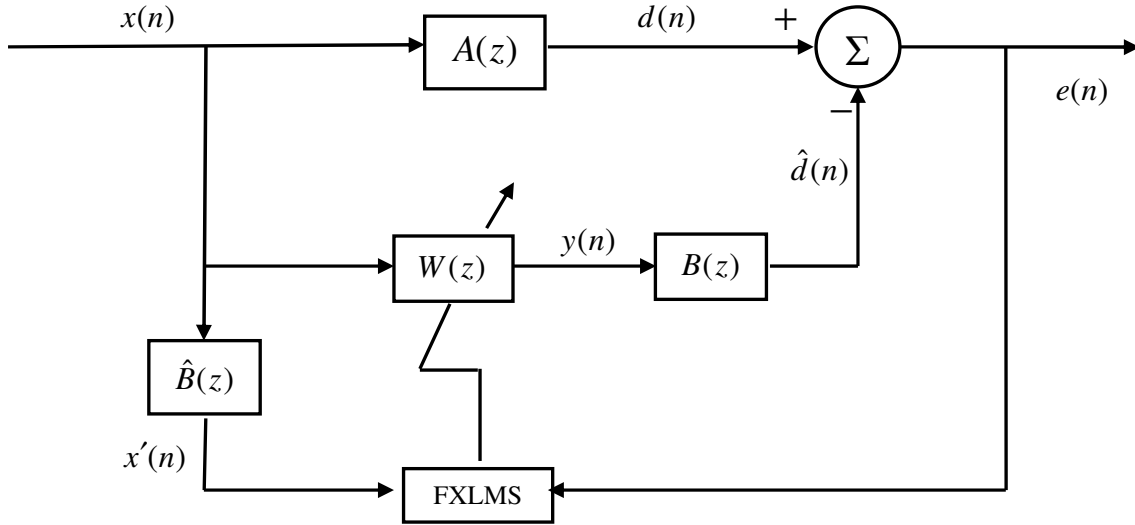


Fig. 2.2 Block diagram of a single-channel feedforward ANC.

The reference noise is distorted by the path while travelling to the secondary loud speaker position. So the characteristic of this path should be analyzed to achieve good results. The basic block diagram representing the scheme of ANC of fig.2.1 is shown in fig.2.2. The path from primary microphone to error sensor is defined as primary path and is denoted by $A(z)$. Similarly the characteristic of the path from secondary loudspeaker to error microphone is also important and should be analyzed thoroughly. This path is defined as secondary path and denoted by $B(z)$. The secondary path also includes the D/A (digital to analog) converter, reconstruction filter, power amplifier, loudspeaker, acoustic path from loudspeaker to error microphone, preamplifier, antialiasing filter, and A/D (analog to digital) converter [1],[10]. Typical primary path impulse response and secondary path impulse response estimated practically are shown in the fig.2.3 and fig.2.4 respectively. While designing the active noise canceller, the presence of primary path

and secondary path must be taken into account as both the paths distort the noise and antinoise signals.

The acoustic noise source and the environment are time varying, the frequency content, amplitude, phase, and velocity of the undesired noise are nonstationary (time varying). So an active noise control system must be adaptive in order to cope with these

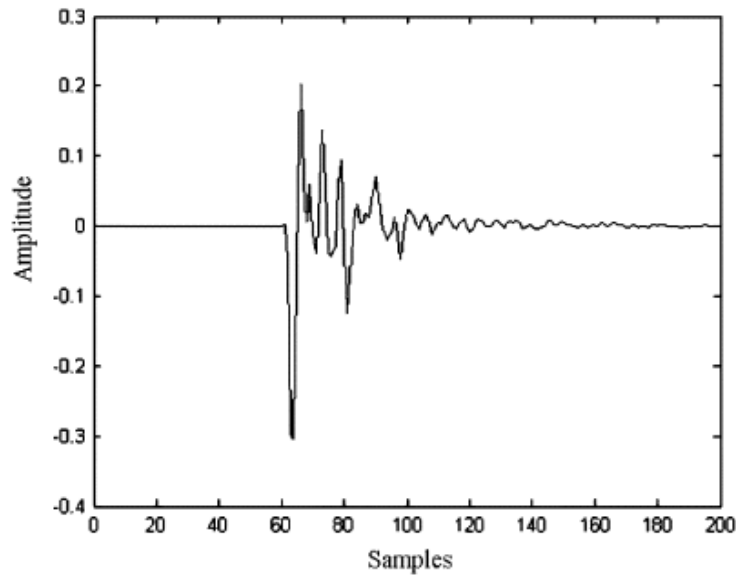


Fig 2.3 Impulse response of primary path (V. DeBrunner and D. Zhou [12]).

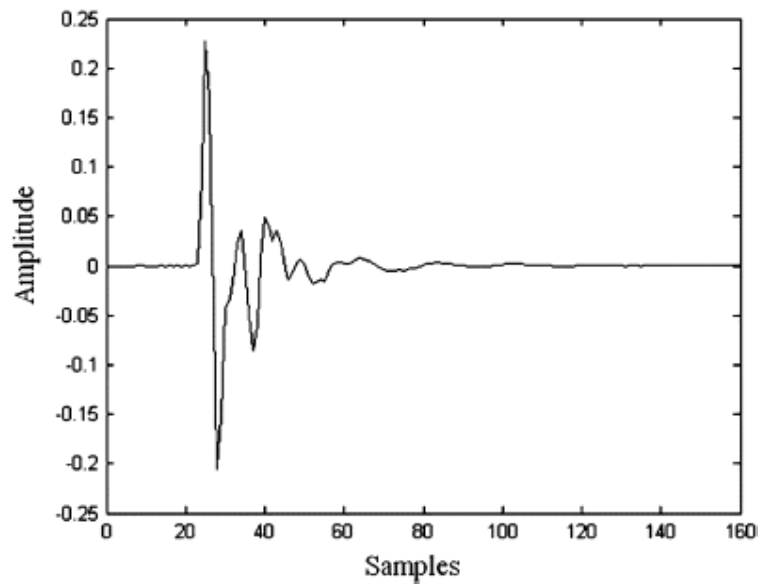


Fig 2.4 Impulse response of Secondary path (V. DeBrunner and D. Zhou [12]).

changing characteristics. This is the reason why modern active noise control systems depend heavily on digital signal processing because in the field of digital signal processing, there are classes of systems called adaptive systems which have the capability to vary their coefficients in order to cope with changing environment. Adaptive systems can be implemented as transversal—finite impulse response (FIR), recursive—infinite impulse response (IIR), lattice filters, transform-domain filters. The coefficients of an adaptive filter are adapted to minimize a predefined cost function which is generally a function of error signal. The process of adaptation is automated by DSP adaptation algorithms.

At first glance it seems quite easy to apply directly already available least mean square (LMS) algorithm to adapt the system to time varying environment. But presence of a secondary path is found to be a hindrance to do this. Direct use of LMS algorithm ignoring the presence of secondary path will affect the performance of ANC severely. This leads to the development of filtered-x LMS (FXLMS) algorithm by Widrow [10].

2.2.1 FXLMS Algorithm

Least mean square (LMS) algorithm would have been sufficient for adaptive filter weight update if there was no secondary path transfer function in the ANC. But as mentioned earlier, the secondary path transfer function, $B(z)$ follows the adaptive filter, so the conventional LMS algorithm must be modified to ensure convergence. The solution to this problem is to place an identical secondary path filter in the reference signal path to the weight update of the LMS algorithm. This realizes the so called filtered-x (FXLMS) algorithm for ANC. The block diagram of a single channel feedforward ANC using FXLMS algorithm is shown in fig.2.2. Referring to fig.2.2, $x(n)$ is the reference noise generated by the noise source at time n . Reference noise travels through the primary path to reach at the cancellation point. So, $d(n)$ is the noise to be cancelled at the cancellation point which is obtained by convolving reference noise with primary path as follows

$$d(n) = \sum_{k=0}^{L_a} x(n-k)a_k(n) \quad (2.1)$$

where $a_k(n)$ is the k^{th} element of the impulse response of the primary path at time n and L_a is the order of primary path. $y(n)$ is the control signal generated by the controller or the adaptive filter, which is given by

$$y(n) = \sum_{k=0}^{N-1} x(n-k)w_k(n) \quad (2.2)$$

where $w_k(n)$ is the k^{th} element of the impulse response of the adaptive filter $\mathbf{w}(n)$ at time n and N is the order of adaptive filter. $\hat{d}(n)$ is the antinoise generated by the ANC via the secondary loudspeaker, which is obtained by filtering the controller signal, $y(n)$, through the secondary path filter

$$\hat{d}(n) = \sum_{k=0}^{M-1} y(n-k)b_k(n) \quad (2.3)$$

where $b_k(n)$ is the k^{th} element of the impulse response of the secondary path $\mathbf{b}(n)$ at time n and M is the order of secondary path. The residual noise or error signal collected by the error microphone after noise cancellation is denoted as $e(n)$. The residual noise is expressed as $e(n)$

$$\begin{aligned} e(n) &= d(n) - \hat{d}(n) \\ &= d(n) - \mathbf{b}(n) * \mathbf{y}(n) \\ &= d(n) - \mathbf{b}(n) * [\mathbf{w}^T(n)\mathbf{x}(n)] \end{aligned} \quad (2.4)$$

where $\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{N-1}(n)]^T$

is the coefficient vector of adaptive filter $W(z)$ at time n , and

$$\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T$$

is the reference signal vector at time n .

The objective of the adaptive filter is to minimize the instantaneous squared error, the cost function, as given below

$$\xi(n) = e^2(n) \quad (2.5)$$

To achieve this, the most widely used method is the stochastic gradient or LMS algorithm which updates the coefficient vector in the negative gradient direction on the MSE surface.

The weight update equation is defined as

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2} \nabla \xi(n) \quad (2.6)$$

where $\nabla \xi(n)$ is an instantaneous estimate of gradient on the MSE surface at time n . This can be expressed as

$$\nabla \xi(n) = \frac{\partial e^2(n)}{\partial \mathbf{w}(n)} = 2e(n) \frac{\partial e(n)}{\partial \mathbf{w}(n)}$$

Putting value of $e(n)$ from (2.4) we obtain

$$\frac{\partial e(n)}{\partial \mathbf{w}(n)} = \frac{\partial [d(n) - \mathbf{b}(n) * \{\mathbf{w}^T(n) \mathbf{x}(n)\}]}{\partial \mathbf{w}(n)}$$

But $d(n)$ is independent of $\mathbf{w}(n)$ so the gradient becomes

$$\frac{\partial e(n)}{\partial \mathbf{w}(n)} = -\mathbf{b}(n) * \mathbf{x}(n) = \mathbf{x}'(n)$$

So $\mathbf{x}'(n)$ is the reference noise signal vector filtered through secondary path filter. Putting the above value in (2.6) we obtain

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{x}'(n) \quad (2.7)$$

where μ is the step size which regulates speed of convergence and stability. This is popularly known as filtered-x LMS (FXLMS) algorithm. This algorithm shows that when secondary path, $B(z)$, follows the adaptive filter, this transfer function must also be placed in the reference signal path. The basic ANC system described above perform quite well in reducing broadband as well as narrowband noise in ducts under plane wave conditions. In this arrangement primary path and secondary path are assumed to be linear in nature and so represented by linear transfer functions. This type of arrangement is called linear ANC

and is adapted by the relatively simple FXLMS algorithm which requires less computation.

To implement FXLMS algorithm the secondary path filter $B(z)$ is to be estimated first. Many offline and online methods are available for identifying the secondary path filter. FXLMS algorithm found to be tolerant to errors made in the estimation of $\hat{B}(z)$. FXLMS algorithm generally converge even with a phase estimation error of upto $\pm 90^\circ$, within the limit of slow adaptation [13].

The maximum allowable step size for FXLMS algorithm is approximately [10]

$$\mu_{\max} = \frac{1}{P_{x'}(N + \Delta)} \quad (2.8)$$

where $P_{x'} = E[x'^2(n)]$ is the mean square value or power of the filtered reference signal $x'(n)$, N is the number of adaptive filter coefficients and Δ is the number of samples corresponding to the overall delay in the secondary path. The delay in the secondary path is the most significant factor influencing the convergence behavior of the ANC system, thus reducing the maximum step size in the FXLMS algorithm.

2.2.2 FELMS Algorithm

E. A. Wan [14] and Popovich [15] introduced an alternative to FXLMS algorithm by developing adjoint LMS algorithm. Instead of filtering the reference input by estimated secondary path (this is pre-estimated before the ANC adaptation), they suggested filtering the error signal by an adjoint secondary path. The block diagram of ANC system using this technique is shown in fig.2.5. The adjoint LMS algorithm is given below

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e'(n)\mathbf{x}(n-M+1) \quad (2.9)$$

where $e'(n)$ is the error signal filtered through the adjoint secondary path filter and M is the order of secondary path filter. Here sufficient number of delays must be added to reference signal to keep the reference and error signals properly time aligned. The number of delays is generally order of secondary path filter. In another method, secondary path equalization (SPE) [1], error signal is filtered through the inverse of secondary path transfer function and delays are provided to reference signal to keep the reference and

error signals time aligned. Both the methods involve filtering the error signal and so the algorithm is called filtered-e LMS (FELMS) algorithm. V. DeBrunner and D. Zhou [12] introduced hybrid filtered error LMS algorithm to further enhance the convergence rate. The FELMS algorithms result in identical residual MSE performance compared to FXLMS algorithm. The computational complexity of the FELMS algorithm is much less compared to FXLMS algorithm particularly for multi-input ANC. However, FELMS algorithm introduces delays to the update step of the algorithm, which slows down the speed of convergence.

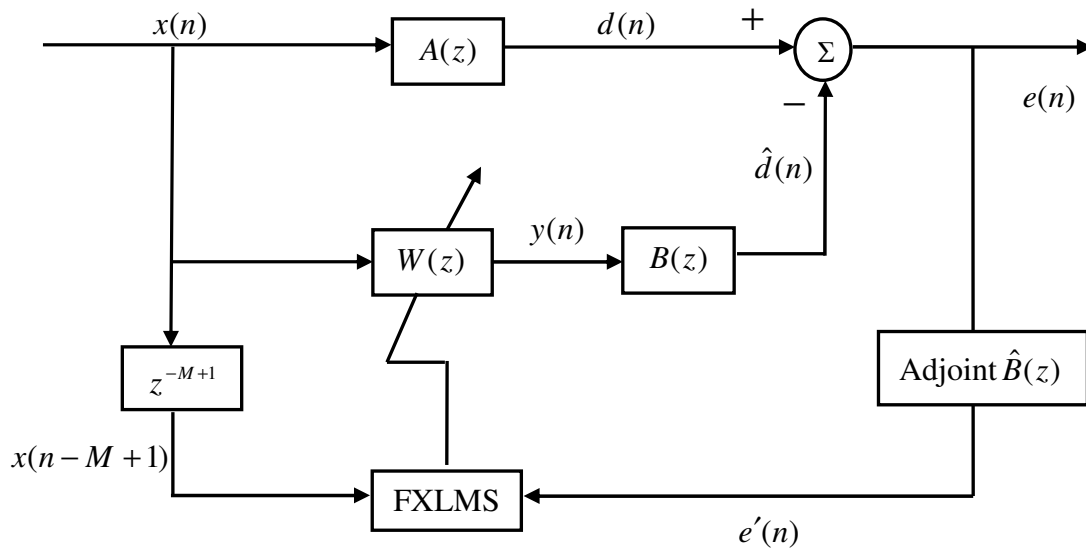


Fig.2.5 Block diagram of ANC using adjoint LMS algorithm.

2.2.3 Feedback Effect

The antinoise generated by the secondary loudspeaker to cancel the reference noise may travel towards the noise source direction. If this antinoise is able to reach the reference microphone, it will collect antinoise along with reference noise signal. This results in a corrupted reference signal $x(n)$. This is called feedback effect which results in potential instability if the gain of this feedback loop becomes too large. Solution to this problem is to use directional microphones and loudspeakers, provide a feedback neutralization filter or use IIR filter.

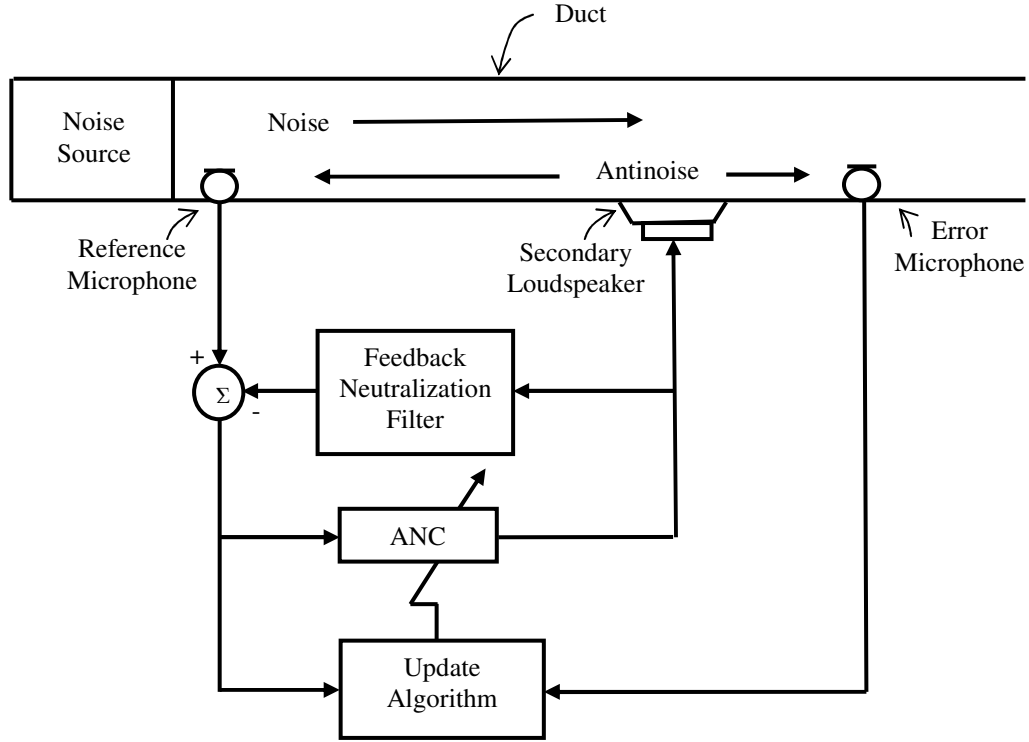


Fig. 2.6 Schematic diagram of ANC with feedback neutralization filter.

The basic arrangement to neutralize feedback effect in a duct by introducing a feedback neutralization filter is shown in fig.2.6. The block diagram representation of the above arrangement is presented in fig. 2.7. Here the feedback component of the reference microphone signal is cancelled by the output of feedback neutralization filter which models the transfer function from secondary loudspeaker input to reference microphone output. The signal actually captured by the reference microphone is the reference noise as well as the feedback signal given by

$$u(n) = x(n) + y(n) * f(n) \quad (2.10)$$

Finally the input signal provided to the ANC is calculated as follows

$$x(n) = u(n) - y(n) * \hat{f}(n) \quad (2.11)$$

where $f(n)$ and $\hat{f}(n)$ are the impulse response of feedback path and estimated feedback path filter respectively. The success of this technique depends on the accuracy of $\hat{F}(z)$ in

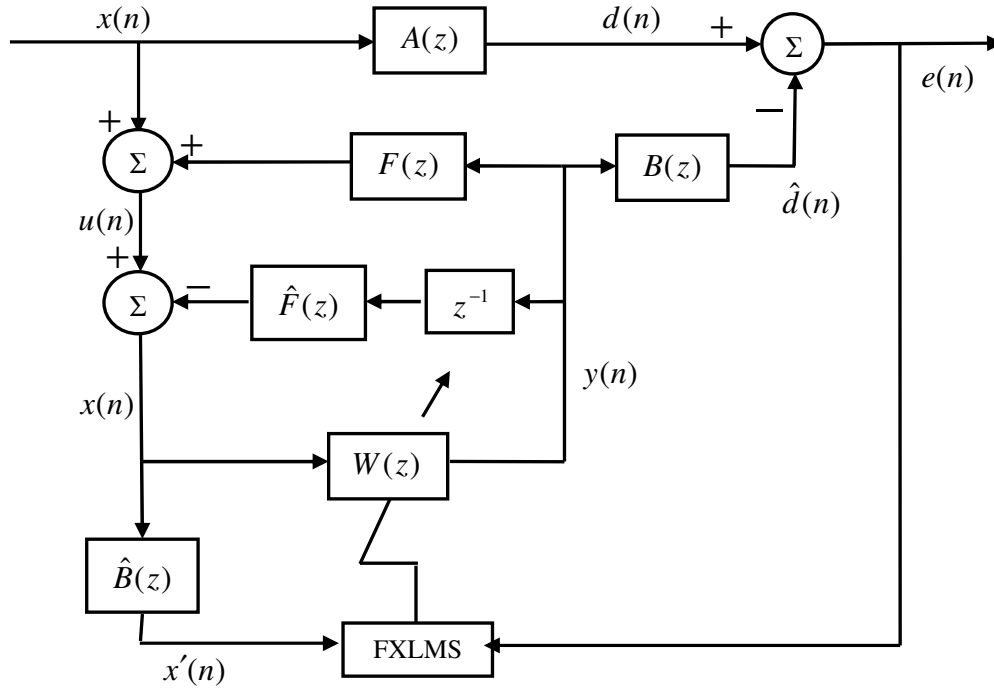


Fig. 2.7 Block diagram of ANC with feedback neutralization filter.

estimating of $F(z)$. The estimation is done offline in the absence of reference noise and is kept fixed during the ANC operation. But in some cases feedback path varies with time. In that case $\hat{F}(z)$ is to be estimated online, along with adaptation of ANC adaptive filter $W(z)$. This creates a problem as $y(n)$ is highly correlated not only with feedback antinoise signal (which is essential for adapting $\hat{F}(z)$) but also with the reference noise (which is essential for adapting $W(z)$). In this situation $\hat{F}(z)$ will continue to adapt even after removing the feedback component from the input signal and so creates problems in the adaptation of $W(z)$ if the convergence rate of $\hat{F}(z)$ and $W(z)$ are comparable. This is the reason why online estimation of feedback path filter is generally avoided.

2.2.4 ANC using IIR Filter

It is well known that IIR adaptive filters have the ability to yield matching characteristics with fewer filter coefficients compared to FIR adaptive filters. Reduction in number of filter coefficients leads to reduction in computational complexity for ANC

implementation. Acoustic feedback in ANC introduces poles in the system. In this situation IIR adaptive filters can better match the physical system as they have zeros as well as poles whereas FIR filters have only zeros.

The dashed box in the fig.2.8 shows the IIR adaptive filter. The output of the adaptive filter is calculated as

$$\mathbf{y}(n) = \mathbf{w}_f^T(n)\mathbf{x}(n) + \mathbf{w}_b^T(n)\mathbf{y}(n-1) \quad (2.12)$$

where $\mathbf{w}_f(n)$ is the weight vector of $W_f(z)$ at time n and $\mathbf{w}_b(n)$ is the weight vector of $W_b(z)$ at time n . $\mathbf{x}(n)$ and $\mathbf{y}(n-1)$ are defined as below

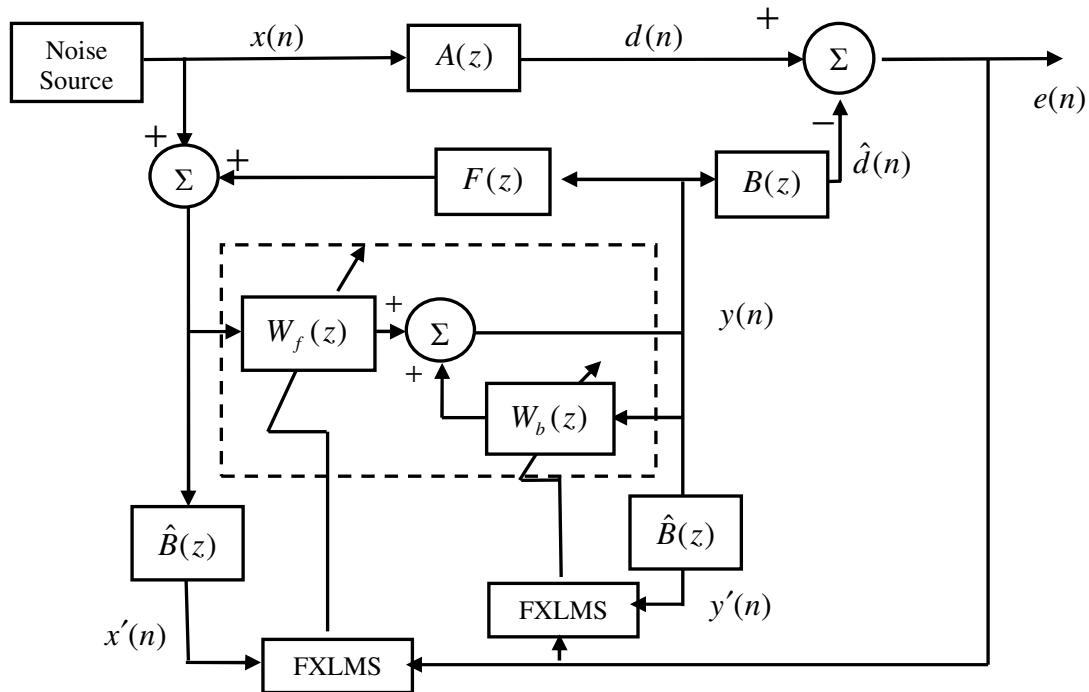


Fig. 2.8 Block diagram of ANC (with feedback) using IIR adaptive filter.

$$\mathbf{x}(n)=[x(n) \ x(n-1) \ \dots \ x(n-L_{w_f}-1)]^T$$

$$\mathbf{y}(n)=[y(n-1) \ y(n-2) \ \dots \ y(n-L_{w_b}-2)]^T$$

L_{w_f} and L_{w_b} are length of weight vector of $W_f(z)$ and $W_b(z)$ respectively.

Define a new overall weight vector

$$\mathbf{w}(n) = \begin{bmatrix} \mathbf{w}_f(n) \\ \mathbf{w}_b(n) \end{bmatrix}$$

And a new reference vector

$$\mathbf{u}(n) = \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{y}(n) \end{bmatrix}$$

Then IIR adaptive filter output can be written as

$$y(n) = \mathbf{w}^T(n)\mathbf{u}(n) \quad (2.13)$$

Now proceeding in the similar line as for the FXLMS algorithm, the weight update equation for IIR adaptive filter can be written as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{u}'(n)e(n) \quad (2.14)$$

where $\mathbf{u}'(n)$ is the $\mathbf{u}(n)$ filtered through the estimated secondary path filter $\hat{B}(z)$. This algorithm can be partitioned to derive two separate weight update equation for $\mathbf{w}_f(n)$ and $\mathbf{w}_b(n)$ as described below

$$\mathbf{w}_f(n+1) = \mathbf{w}_f(n) + \mu \mathbf{x}'(n)e(n) \quad (2.15)$$

$$\mathbf{w}_b(n+1) = \mathbf{w}_b(n) + \mu \mathbf{y}'(n-1)e(n) \quad (2.16)$$

where $\mathbf{x}'(n) = \mathbf{b}(n) * \mathbf{x}(n)$ and $\mathbf{y}'(n) = \mathbf{b}(n) * \mathbf{y}(n-1)$

This algorithm for IIR adaptive filter is called filtered-u recursive LMS algorithm. The only drawback of this algorithm is that even though experimentally it works well, the stability and global convergence is not guaranteed.

2.2.5 Narrowband Feedforward Control

In applications where the reference noise is produced by rotating or reciprocating machines, reference noise is generally periodic (or nearly periodic). This refers to periodic noises generated by engines, compressors, motors, fans, and propellers. In this case, direct observation of the mechanical motion (such as speed) of such noise sources is

possible by using an appropriate sensor. So, here reference microphone can be replaced by a nonacoustic sensor (such as tachometer). This sensor provides an electrical reference signal with the same fundamental frequency as the reference noise emitted. Because all of the repetitive noise occurs at harmonics of the machine's basic rotational frequency, the control system can model these known noise frequencies and generate the antinoise signal. This type of control system is desirable in a vehicle cabin, because it control engine noise but will not affect vehicle warning signals, radio signals or speech, which are not normally synchronized with the engine rotation.

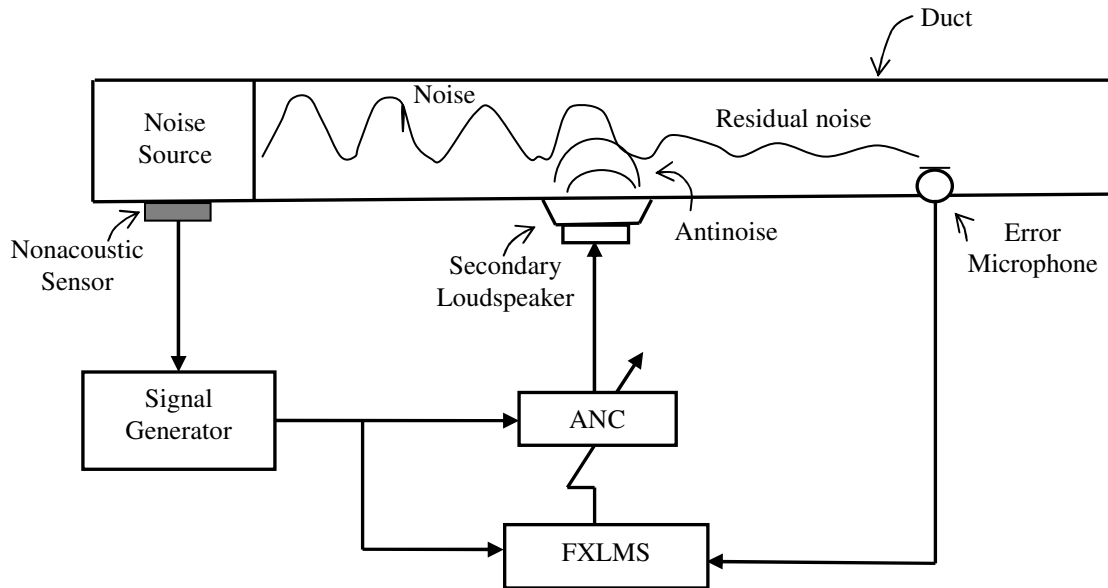


Fig. 2.9 Schematic diagram of narrowband feedforward ANC.

Unfortunately the antinoise produced by secondary loudspeaker propagate in both downstream and upstream direction. So antinoise not only cancels the reference noise but also radiates upstream and reach the reference microphone resulting in a corrupted reference signal. This effect is called feedback effect which introduces poles in the response of the model and thus results in potential instability if the gain of this feedback loop becomes too large. The principal advantage of narrowband feedforward ANC systems is due to the use of nonacoustic sensor (e.g. tachometer) to generate reference input, feedback effect is eliminated. Fig. 2.9 depicts the schematic diagram of a narrowband ANC system.

2.3 Feedback Control

Olson and May [6] first proposed feedback ANC system which uses a carefully designed amplifier matched to the response of error microphone and secondary loudspeaker. The feedback ANC systems use only an error microphone and the active noise controller try to control the noise without having any knowledge about the upstream reference input. Since feedback ANC uses only an error microphone, it also avoids the secondary-to-reference feedback problem inherent in many broadband feedfo-

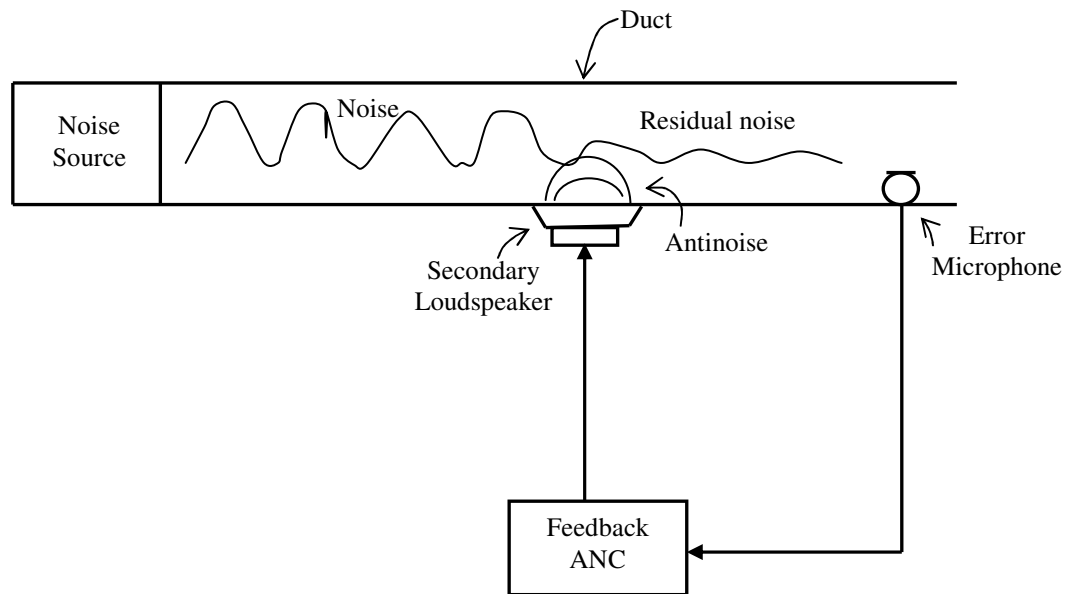


Fig.2.10 Schematic diagram of feedback ANC.

ward ANC systems. Schematic diagram of feedback ANC is shown in the fig. 2.10. Feedforward active noise control is found to be robust and stable in comparison to feedback active noise control [1].

2.4 Multichannel Active Noise Control

Controlling noise inside an enclosure or a large dimension duct is difficult as the noise field is complicated. Another example is transformer noise cancellation. Use of one reference microphone, one secondary loudspeaker and one error microphone is not enough to control such type of complicated noise pattern. So multiple number of reference microphones, secondary loudspeakers and error microphones are employed which is called multiple channel ANC. Multiple channel ANC has large number of primary paths, secondary paths and adaptive filters. All the adaptive filters are generally

adapted independently. Multiple error FXLMS algorithm developed by Elliott and colleagues is generally used for multiple channel ANC [1][2]. Schematic diagram of a multiple channel ANC for an enclosure is shown in the fig. 2.11.

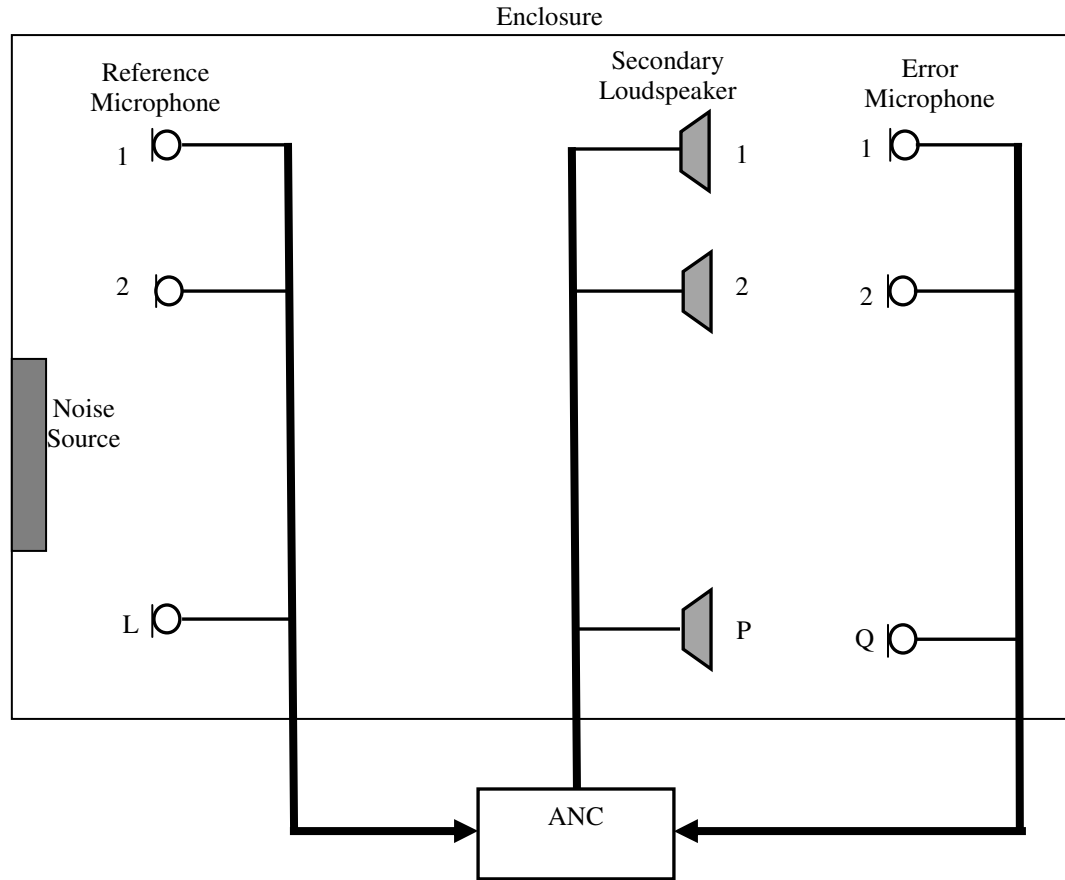


Fig. 2.11 Schematic diagram of multi-channel ANC for an enclosure.

2.5 Virtual Sensor Technique (VST)

All the ANC systems analyzed above can be put within one bracket of the so-called local active noise control system. Local ANC puts all its effort to minimize undesired noise near the error microphone (physical sensor). While this results in a small localized zone of silence being created around the error microphone, the surrounding zone of silence is quite small. Additionally, the noise level outside the zone of silence is likely to be higher than the original noise alone. Virtual acoustic sensors have been developed to overcome the problems associated with local ANC. Virtual acoustic sensors shift the zone of quiet to a desired location that is remote from the error microphone, as shown in fig. 2.12 [16].

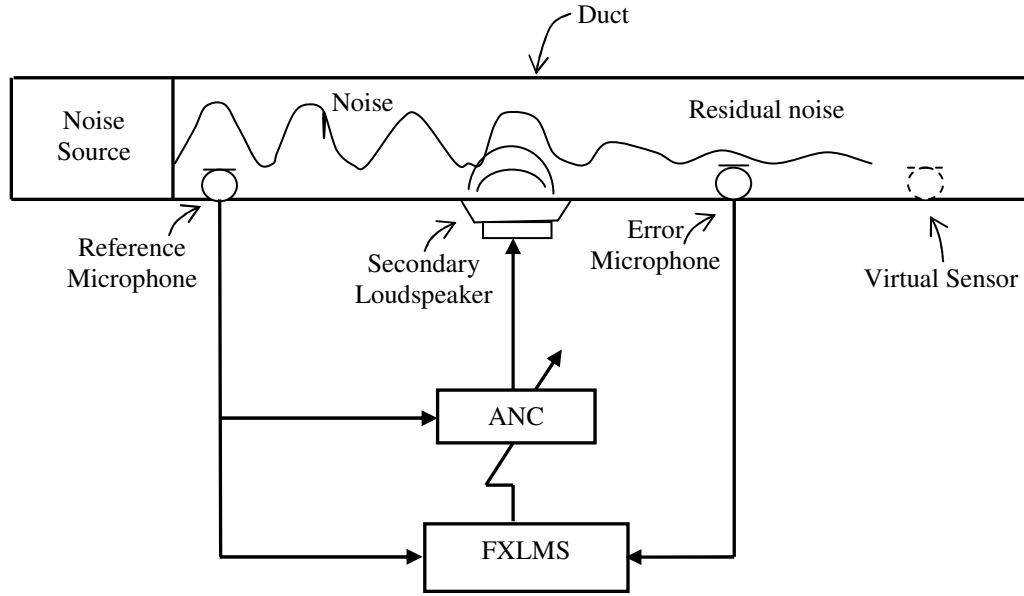


Fig. 2.12 Schematic diagram of ANC using virtual sensor technique.

In this figure, the zone of silence has been moved from the error microphone to the virtual sensor region. The block diagram for the above scheme is shown in fig. 2.13. This technique is also reported as remote microphone technique (RMT) in [17]. In virtual sensor technique the error signal at the virtual location, $\hat{e}_v(n)$, is estimated using the error signal available from the error microphone (physical microphone), $e_p(n)$. To accomplish this task virtual sensor technique requires a preliminary identification stage in which a second physical microphone is temporarily placed at the virtual location. The secondary path transfer functions at the physical and virtual locations, $\hat{B}_p(z)$ and $\hat{B}_v(z)$ respectively, are estimated during the preliminary identification stage along with the primary path transfer function between the physical and virtual locations, $\hat{A}_v(z)$. A block diagram of the virtual sensor technique is given within the dotted box of fig. 2.13. Referring fig. 2.13, first an estimate of the primary disturbance, $\hat{d}_p(n)$, at the physical microphone is calculated using

$$\hat{d}_p(n) = e_p(n) - y_p(n) \quad (2.17)$$

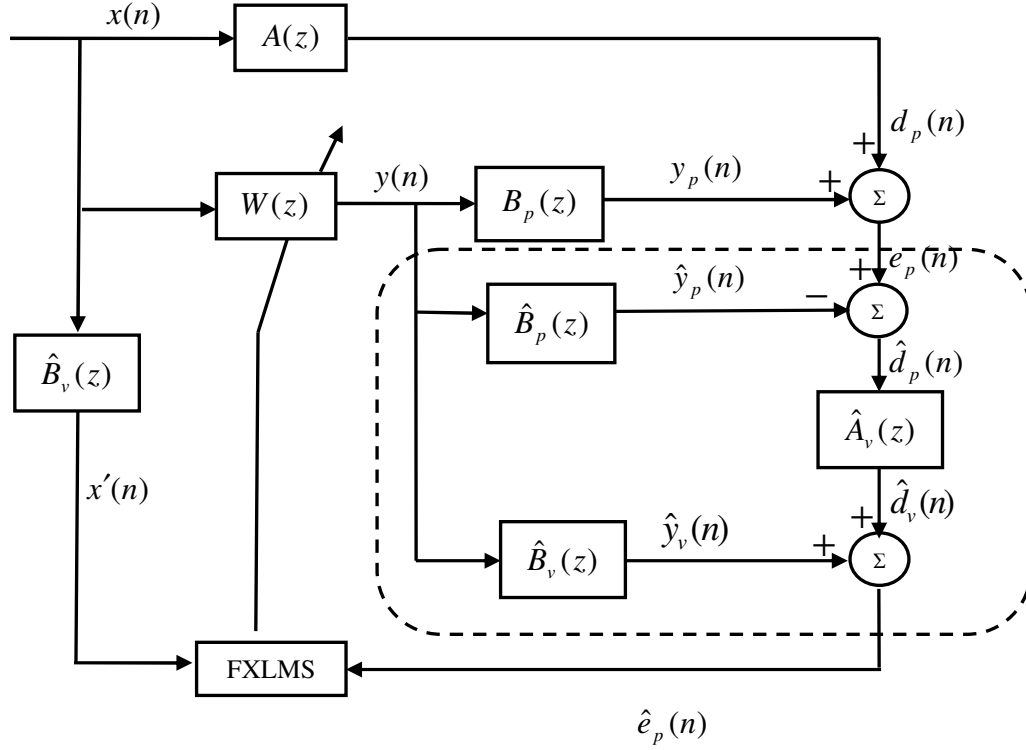


Fig. 2.13 Block diagram of ANC using virtual sensor technique.

where $e_p(n)$ is the actual error signal measured by the error microphone (physical sensor) and $y_p(n)$ is obtained by filtering $y(n)$ through the estimated secondary path at physical microphone, $\hat{B}_p(z)$. Then an estimate of the primary disturbance, $\hat{d}_p(n)$, at the virtual sensor is calculated as follows

$$\hat{d}_p(n) = \hat{a}_v(n) * \hat{d}_p(n) \quad (2.18)$$

where $\hat{a}_v(n)$ is the filter impulse response corresponding to the transfer function $\hat{A}_v(z)$. Finally an estimate of the virtual error signal at the virtual sensor position is calculated as follows (this can also be verified from the block diagram)

$$e_v(n) = \hat{d}_v(n) + \hat{y}_v(n) \quad (2.19)$$

where $\hat{y}_v(n)$ is the estimated secondary disturbance at the virtual sensor position. Thus an estimate of the virtual error signal has been calculated from the physical error signal. The

adaptive filter having transfer function $W(z)$ can be updated using the traditional FXLMS algorithm as below

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e_v(n) \mathbf{x}'(n) \quad (2.20)$$

where $\mathbf{x}'(n)$ is reference input filtered through $\hat{B}_v(z)$.

2.6 Frequency Domain Approach

Frequency domain adaptive filters have three major advantages over time-domain adaptive filters

- (i) The potential saving in the computation by using Fast Fourier Transform (FFT)
- (ii) More accurate estimation of the gradient due to the averaging of the samples in the whole data block and
- (iii) Rapid convergence by using normalized step sizes for each frequency bin

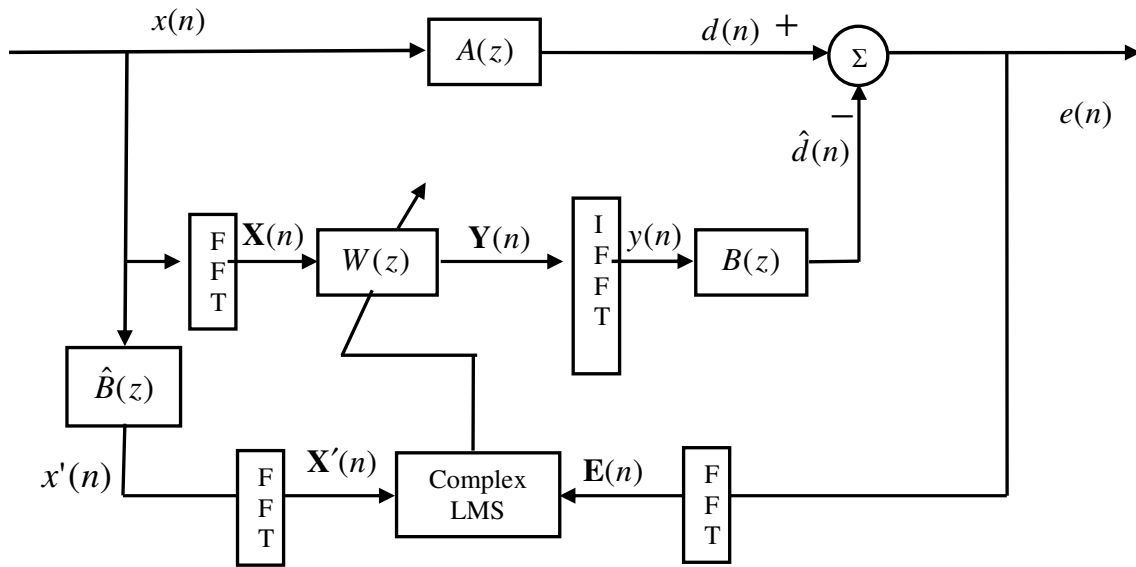


Fig.2.14 Block diagram of ANC using frequency-domain FXLMS algorithm.

ANC can be implemented in the frequency domain to take benefits of the above mentioned advantages. Block diagram for frequency domain implementation of ANC is shown in the fig.2.14. A block of reference signal samples are transformed to the frequency domain signal $\mathbf{X}(n)$ using FFT. $\mathbf{X}(n)$ is then filtered through the frequency

domain adaptive filter $W(z)$ to produce frequency domain output $\mathbf{Y}(n)$. Then by IFFT of $\mathbf{Y}(n)$ we get the time-domain output signal vector $y(n)$. The residual error $e(n)$ and filtered reference signal $x'(n)$ are also transformed to frequency-domain by FFT to get $\mathbf{E}(n)$ and $\mathbf{X}'(n)$ respectively. Now the adaptive filter coefficients are updated by the complex LMS algorithm expressed as [1]

$$\mathbf{W}(n+N) = \mathbf{W}(n) + \mu \mathbf{X}'(n)\mathbf{E}(n) \quad (2.21)$$

where N is the input signal block length. This is called frequency-domain FXLMS algorithm. Instead of filtering the signal sample by sample, frequency-domain FXLMS algorithm processed the signal block by block. This create N sample of delay in the adaptation process causing difficulty in controlling broadband random noise because of the causality constraint. But for periodic reference noise, the effect of block processing delay can be tolerated.

2.7 Active Noise Control in Headset

Headset is generally used for blocking outside noise reaching the ear in highly noisy environment such as flying aircraft or in a subway train. Headphone is also used for listening prerecorded music or voice communication in aviation or military. But in case of a highly noisy environment, outside noise leakage corrupt the desired signal. ANC can be used to cancel the leakage outside noise in headset. In H. F. Olson and E. G. May [6] system a microphone is placed to collect noise and loudspeaker is used to generate antinoise. Shiang-Hwua Yu and Jwu-Sheng Hu [18], Ying Song, Yu Gong, and Sen M. Kuo [19], Thomas Schumacher, Hauke Krüger, Marco Jeub, Peter Vary, Christophe Beaugeant [20] and Cheng-Yuan Chang and Sheng-Ting Li [21] developed ANC headset basically using feedback control. Huge change in the primary path and secondary path is observed when the position of the headset is changed and possesses the real challenge in this application. Romain Serizel, Marc Moonen, Jan Wouters, and Søren Holdt Jensen [22] and Romain Serizel, Marc Moonen, Jan Wouters, and Søren Holdt Jensen [23] use ANC technique in hearing aid where as W. S. Gan and S. Kuo [24] developed a headphone having an integrated audio and active noise controller. Here the headphone speaker playing the music also generates antinoise to cancel the outside leakage noise.

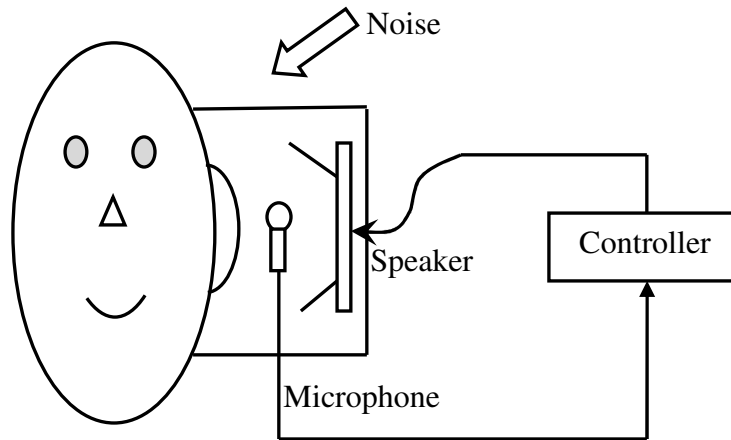


Fig. 2.15 Schematic diagram of ANC used for headphone or headset.

2.8 ANC in functional Magnetic Resonance Imaging (fMRI)

Magnetic Resonance Imaging (MRI) was widely used in medical and clinical researches during the past 20 years. functional MRI (fMRI) has also been applied in neuroscience and psychology studies. High speed echo planar imaging (EPI) technique is usually applied in fMRI study to acquire high temporal resolution signals [25]. However, fast switch of magnetic gradients during EPI acquisition induced loud acoustic noise, which will up to 100 dBL or even louder in higher magnetic field. This acoustic noise can interfere in the communication between staff and volunteers or patients, impair the hearing ability, and suppress brain activations in fMRI study. People usually used passive component, such as earmuff or earplug, to prevent hearing damage from the MRI noise. However, the noise reduction was limited by the design and the material of the passive components and the passive components can also hardly solve the communication problem between staff and volunteers. The other choice for reducing the MRI noise is active noise cancellation (ANC) system. Active Noise Control (ANC) of fMRI acoustic noise using the conventional FXLMS approach results in poor cancelation performance and slow convergence due to its broadband nature and the need for high order adaptive filters. High order adaptive filters are needed to effectively model the long acoustic impulse responses. Existing methods to improve the performance of FXLMS based broadband ANC systems are either computationally expensive or need elaborate implementation. Casper K. Chen, Tzi-Dar Chiueh and Jyh-Horng Chen [25] introduced a

new neural-network architecture for reducing the acoustic noise level in magnetic resonance (MR) imaging processes. The proposed neural network (NN) consists of two cascaded time-delay NN's (TDNN's). This NN is used as the predictor of a feedback active noise control (ANC) system for reducing acoustic noises. Experimental results with real MR noises show that the proposed system achieved an average noise power attenuation of 18.75 dB. Kuan-Hung Cho, Tzi-Dar Chiueh, Ching-Po Lin, Casper K. Chen, Jyh-Horng Chen [26] experimented an ANC for fMRI using an IIR filter. They have showed that the ANC system could provide extra 9.4 dB reduction at 937 Hz, which is the main frequency of the EPI noise. Besides, the fMRI results showed that there was apparent enhancement in both activated pixel numbers and the activation strength in visual cortex (VC). Govind Kannan, Ali A. Milani, Issa M.S. Panahi, Nasser Kehtarnavaz [27] developed Wiener initialized FXLMS (WI-FXLMS) algorithm and demonstrated the effectiveness of this approach for the active noise control of functional MRI acoustic noise and several other realistic noise sources.

2.9 ANC in Infant Incubators

Intense level of noise inside infant incubators like neonatal intensive care unit (NICU) is due to the operation of various medical instruments. These loud noises can cause serious psychological effects in infants such as changes in heart rate, blood pressure, oxygenation, respiration, intestinal peristalsis, and glucose consumption. Lichuan Liu, Shruthi Gujjula and Sen M. Kuo [29] used ANC to reduce this noise. They have done real time experiments with single channel, multi channel and pseudo-multi channel ANC. Multi channel ANC, $1 \times 2 \times 2$ (1 reference microphone, 2 secondary speaker and 2 error microphone), yield better noise reduction results but require more computational complexity which is a problem for real time implementation. They proposed pseudo multi channel ANC with same number of sensors as in multichannel ANC ($1 \times 2 \times 2$). Here after the digital output $y(n)$ is converted to analog signal, it is amplified to drive both loudspeakers and two analog signals picked up by two error microphones are mixed by an analog mixer to get an single analog error signal. So here computational complexity remains same as single channel ANC.

Other Applications of ANC

ANC has been used for other applications like transformer noise cancellation by X. Qiu and C. H. Hansen [30] and vehicle interior noise by Ji-guang Jiang , Deng-feng Wang, Yue Zeng, Jun-ting Wang, Xiao-lin Cao [31].

2.10 Study on Nonlinear ANC

The ANC systems discussed till now use an adaptive filter which is either a finite impulse response (FIR) filter or an infinite impulse response (IIR) filter. The adaptive algorithm derived for ANC using FIR adaptive filter is FXLMS algorithm. The ANC using IIR structure as the adaptive filter is based on filtered-u least mean square (FULMS) algorithm. Both the filter structures, FIR and IIR filter, have a linear form in the sense that the output has a linear relationship with filter coefficients and reference input. This enable the adaptive filters, using either FIR or IIR structure, to perform noise reduction when the primary path and secondary paths are linear. But real time implementation reveals that nonlinearity exists in ANC. Some of the areas which lead to the appearance of nonlinearity in the ANC are listed below

- The reference noise at the cancellation point may exhibit nonlinear distortion due to the nonlinear transfer function of the primary path, for example, primary noise propagating in a duct with very high sound pressure.
- The secondary path between the speaker and the error microphone may exhibit nonlinear behavior, for example when the amplitude of the antinoise is greater than the saturation limit of the speaker or the frequency of the antinoise is less than the cutoff frequency of the speaker.
- The reference noise may itself exhibit nonlinearity. The noise from a dynamic system may be a nonlinear and deterministic (chaotic rather than stochastic, white, or tonal) noise process. For example, fan noise often shows chaotic behavior.
- The reference and error sensors may be saturated in real world applications if the noise level exceeds the dynamic range of the sensors. The loudspeaker, for example, can excite both the frequency of interest and its associated harmonics.
- Nonlinearity arising out of specific applications (personal hearing aid, long duct).

Careful examination of a practical ANC system revealed that the nonlinearity must be taken into account in order to achieve better overall performance. This analysis paved the way for the development of a new class of nonlinearity tolerated ANC henceforth called nonlinear active noise control (NANC) system.

Nonlinearity sneaks into the ANC systems because of one or more or combination of above mentioned factors. This leads to a class of new ANC better known to the researchers as nonlinear active noise control (NANC). Under this circumstances if the existing ANC techniques (which use FIR or IIR filter as the adaptive filter) discussed earlier are employed, performance degradation is noticed. The obvious reason being application of a linear technique for an ANC which is actually nonlinear in nature. The solution to this problem is to explore various available nonlinear filtering techniques. These methods need proper analysis and be suitably modified to apply conveniently to the problem of nonlinear ANC. The following methods reported in the literature have been applied to address the issue of nonlinear ANC.

- Adaptive Volterra Filter (AVF)
- Adaptive Bilinear Filter (ABF)
- Mutli-Layer Artificial Neural Network (MLANN)
- Radial Basis Function (RBF)
- Recurrent Neural Network (RNN)
- Fuzzy Logic (FL)
- Fuzzy Neural Network (FNN)
- Functional Link Artificial Neural Network (FLANN)
- Genetic Algorithm (GA)
- Particle Swarm Optimization (PSO)

While application of MLANN and FLANN have been analyzed in detail in the subsequent chapters, other methods mentioned above are discussed briefly here. Different variations of the above mentioned techniques exist. The variations generally include modifications in the update algorithm. However, none of them provide global minimum so far as the error minimization is concerned in the sense that if one works well for a given application fails to achieve the same in the other application. These methods can

be evaluated based under the following characteristics embedded into the algorithm
 (i) Mean Square Error (MSE) achieved after convergence (ii) Speed of convergence (iii) Computational complexity requirement (iv) Stability

2.10.1 Adaptive Volterra Filter (AVF) Approach

In [32][33] V. John Mathews explains how Volterra series expansion can be used to build adaptive filters to deal with polynomial model of nonlinearity. This is found to be very useful for practical applications which involve nonlinear processing of signals. Filters involving Volterra series expansion are attractive in adaptive filtering applications because the expansion is a linear combination of nonlinear functions of the input signal. In another way, the output of this filter is linear with respect to filter coefficients.

Li-Zhe Tan and Jean Jiang [34] reported an adaptive filtered-x algorithm for ANC using second order Volterra filter. The developed algorithms can be used as alternatives in the case where the standard filtered-x LMS algorithm does not perform well e.g. for nonlinear ANC. The block diagram of ANC using second order Volterra filter is shown in the fig. 2.16.

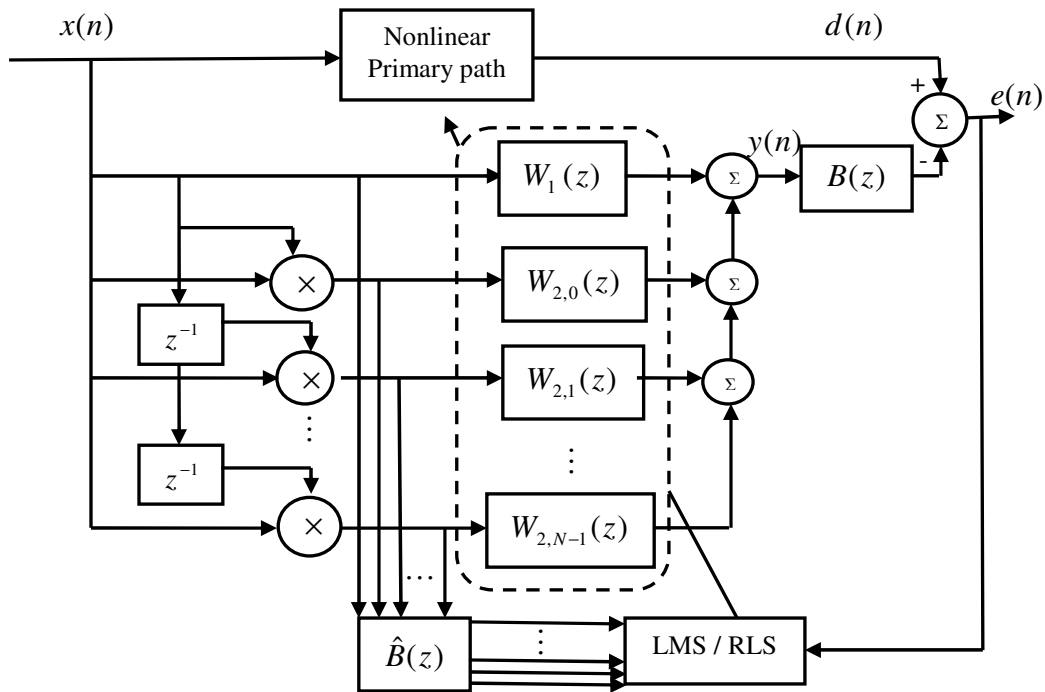


Fig.2.16 Block diagram of ANC using filtered-x second order Volterra adaptive algorithm.

Consider a second order Volterra filter described by the following input and output relationship

$$y(n) = \sum_{i=0}^{N-1} w_1(i, n)x(n-i) + \sum_{i=0}^{N-1} w_{2,0}(i, n)x^2(n-i) + \sum_{i=0}^{N-2} w_{2,1}(i, n)x(n-i)x(n-1-i) + \dots + w_{2,N-1}(0, n)x(n-N+1) \quad (2.22)$$

where $x(n)$ and $y(n)$ represents the filter input and output respectively, N is the memory span and $w_1(i, n), w_{2,0}(i, n), w_{2,1}(i, n), \dots, w_{2,N-1}(0, n)$ are filter coefficients. The above equation is implemented as a number of filters operating in parallel. Unlike FXLMS algorithm where one adaptive FIR filter is used this method use $N+1$ number of FIR adaptive filters. The coefficient vector of the adaptive filters involved are defined below

$$\begin{aligned} \mathbf{w}_1(n) &= [w_1(0, n), w_1(1, n), \dots, w_1(N-1, n)] \\ \mathbf{w}_{2,0}(n) &= [w_{2,0}(0, n), w_{2,0}(1, n), \dots, w_{2,0}(N-1, n)] \\ \mathbf{w}_{2,1}(n) &= [w_{2,1}(0, n), w_{2,1}(1, n), \dots, w_{2,1}(N-2, n)] \\ \mathbf{w}_{2,2}(n) &= [w_{2,2}(0, n), w_{2,2}(1, n), \dots, w_{2,2}(N-3, n)] \\ &\dots\dots\dots \\ \mathbf{w}_{2,N-1}(n) &= [w_{2,N-1}(0, n)] \end{aligned}$$

The order of first filter is N , next starting from second filter the filter order reduces from N to 1.

The input to the adaptive filters are defined below

$$\begin{aligned} \mathbf{x}_1(n) &= [x(n), x(n-1), \dots, x(n-N+1)] \\ \mathbf{x}_{2,0}(n) &= [x^2(n), x^2(n-1), \dots, x^2(n-N+1)] \\ \mathbf{x}_{2,1}(n) &= [x(n)x(n-1), x(n-1)x(n-2), \dots, x(n-N+2)x(n-N+1)] \\ \mathbf{x}_{2,2}(n) &= [x(n)x(n-2), x(n-1)x(n-3), \dots, x(n-N+3)x(n-N+1)] \\ &\dots\dots\dots \\ \mathbf{x}_{2,N-1}(n) &= [x(n)x(n-N+1)] \end{aligned}$$

The adaptive filters output are obtained by filtering the inputs through the corresponding adaptive filter and then are added to get the output of the controller as follows

$$y(n) = y_1(n) + y_{2,0} + \dots + y_{2,N-1} \quad (2.23)$$

As shown in the figure residual error signal is given by

$$e(n) = d(n) - \mathbf{b}(n) * y(n) \quad (2.24)$$

where $\mathbf{b}(n)$ is the impulse response of the secondary path transfer function $B(z)$.

Tan and Jiang [34] used the standard FXLMS algorithm to develop filtered-x second order Volterra LMS algorithm and update the adaptive filter. Similar to FIR filter here the goal of the adaptive algorithm is to minimize the instantaneous square error using the steepest descent algorithm. Weight update formula for the adaptive filters are written as follows

$$\begin{aligned}\mathbf{w}_1(n+1) &= \mathbf{w}_1(n) + \mu_1 e(n) \mathbf{x}'_1(n) \\ \mathbf{w}_{2,i}(n+1) &= \mathbf{w}_{2,i}(n) + \mu_2 e(n) \mathbf{x}'_{2,i}(n)\end{aligned}\quad (2.25)$$

where $i=0,1,2,\dots,N-1$, μ_1 is the step size for first adaptive filter and μ_2 is the step size for all other adaptive filters.

Here it can be noticed that the input to first adaptive filter is $x(n)$ with its delayed samples. The input to second adaptive filter is $x^2(n)$ with its delayed samples (squared terms) and to all other adaptive filters are $x(n)x(n-1), x(n)x(n-2)$ etc. with their delayed samples (cross-terms). So the first adaptive filter represents linearity of the ANC where as all other filters represent nonlinearity of the system.

They have also used FXRLS algorithm to develop filtered-x second order Volterra RLS algorithm as follows

$$\mathbf{k}(n) = \frac{\lambda^{-1} P(n-1) \mathbf{x}'_1(n)}{1 + \lambda^{-1} \mathbf{x}'_1(n) P(n-1) \mathbf{x}_1'^T(n)} \quad (2.26)$$

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \mathbf{k}(n) e(n) \quad (2.27)$$

$$P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n) \mathbf{x}'_1(n) P(n-1) \quad (2.28)$$

where $\lambda (0 < \lambda \leq 1)$ is the forgetting factor, $\mathbf{k}(n)$ is the gain vector and \mathbf{x}'_1 is the filtered input signal for the first adaptive filter. Similarly update equations for other adaptive filters are given below

$$\mathbf{k}(n) = \frac{\lambda^{-1} P(n-1) \mathbf{x}'_{2,i}(n)}{1 + \lambda^{-1} \mathbf{x}'_{2,i}(n) P(n-1) \mathbf{x}_{2,i}'^T(n)} \quad (2.29)$$

$$\mathbf{w}_{2,i}(n+1) = \mathbf{w}_{2,i}(n) + \mathbf{k}(n) e(n) \quad (2.30)$$

$$P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} \mathbf{k}(n) \mathbf{x}'_{2,i}(n) P(n-1) \quad (2.31)$$

The algorithms gave better results in terms of MSE for nonlinear ANC. The filtered-x second order Volterra RLS algorithm shows better convergence speed than its LMS counterpart.

Subsequently, Li-Zhe Tan and Jean Jiang [35] reported another paper giving exhaustive theoretical analysis of the above algorithm and named it as Volterra FXLMS (VFXLMS) algorithm. Here the analysis is not limited to second order Volterra filters but higher order filters are included. Inclusion of higher order Volterra filters obviously increases the number of adaptive filters which may enhance the MSE performance of the ANC but raises the computational complexity of the algorithm substantially. They implemented the algorithm in multichannel structure. This structure can be called as filter bank structure [36] in order to avoid confusion with multichannel ANC. They have shown by computer simulation that VFXLMS algorithm gives better MSE result compared to the FXLMS in the following two situations

- 1) The reference noise sensed by a reference microphone is a nonlinear and predictable noise process, while the secondary path transfer function of an ANC system has nonminimum phase.
- 2) The primary path exhibits nonlinear behavior.

2.10.2 Adaptive Bilinear Filter (ABF) Approach

In [32], [33] V. John Mathews and G. L. Sicuranza discussed that Bilinear filters is a popular class of adaptive filter described by the following input and output relationship

$$y(n) = \sum_{i=0}^{N-1} w_1(i, n)x(n-i) + \sum_{i=1}^{N-1} w_2(i, n)y(n-i) + \sum_{i=0}^{N-1} \sum_{j=1}^{N-1} w_{i,j}(n)y(n-j)x(n-i) \quad (2.32)$$

where the filter coefficients $w_1(i, n)$, $w_2(i, n)$ and $w_{i,j}(n)$ can have different length but for simplicity they are assumed to have equal length, N . $x(n)$ and $y(n)$ are input and output of the filter. $w_1(i, n)$ are the feedforward filter coefficients corresponding to input $x(n)$, $w_2(i, n)$ are feedback filter coefficients corresponding to output $y(n)$ and $w_{i,j}(n)$ are cross filter coefficients corresponding to cross terms $x(n)y(n-1)$. In spite of the simplicity, this is an important nonlinear model since a large class of nonlinear systems

can be approximated with arbitrary precision using bilinear system models with finite number of filter weights.

In [37], [38] Sen M. Kuo and Hsien-Tsai Wu used adaptive bilinear filters for nonlinear ANC. There are two types of adaptive bilinear filters: equation-error methods and output-error methods [32], [37], [38]. The output-error algorithm computes the filter output using a truly recursive model since the output signal $y(n)$ is fed back to generate the adaptive estimate of the desired signal $\hat{d}(n)$. The equation-error algorithm calculates the output signal using the input signal $x(n)$ and the desired signal $d(n)$, thus is not a truly recursive estimator. In addition, the equation error method results in biased steady-state solutions, whereas the output-error method provides unbiased estimates. Furthermore, the desired signal $d(n)$, is not available for real-time ANC systems in practical applications, thus the equation-error filter cannot be directly used in them. Therefore they used the adaptive output error bilinear filter for ANC systems. The block diagram of ANC using adaptive bilinear filter is shown in the fig. 2.17, below.

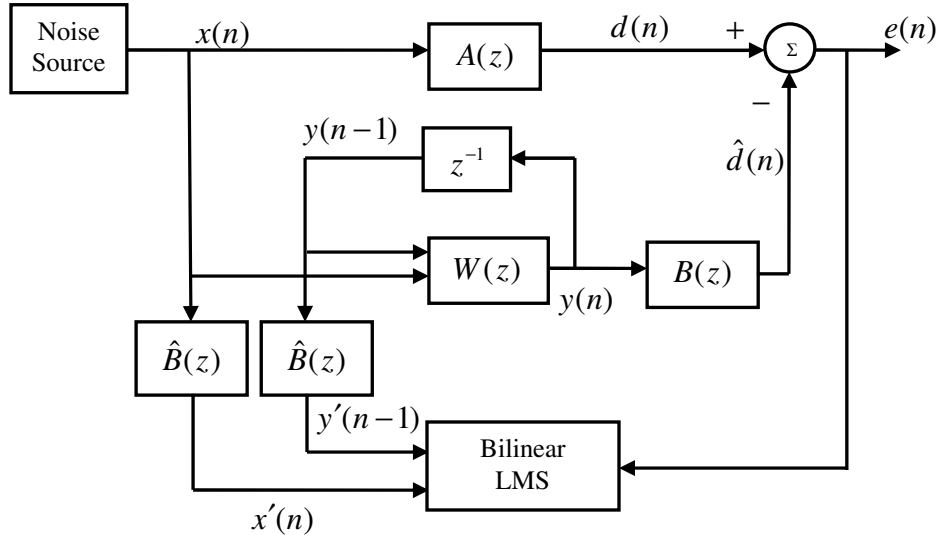


Fig. 2.17 Block diagram of ANC using the adaptive output-error bilinear filter.

The bilinear filter coefficients can be rearranged to form the following vectors.

Feedforward filter coefficient vector $\mathbf{w}_1(n) = [w_1(0, n), w_1(1, n), \dots, w_1(N-1, n)]$

Feedback filter coefficient vector $\mathbf{w}_2(n) = [w_2(0, n), w_2(1, n), \dots, w_2(N-1, n)]$

Cross filter coefficient vector $\mathbf{w}_{i,j}(n) = [w_{0,1}(n), w_{0,2}(n), \dots, w_{N-1,N-1}(n)]$

All the filter coefficient vectors can be combined to form a single vector

$$\mathbf{w}(n) = [\mathbf{w}_1(n), \mathbf{w}_2(n), \dots, \mathbf{w}_{i,j}(n)]$$

The input vector for the three filter coefficient vectors can also be rearranged to form the following vectors

$$\text{i) } \mathbf{x}_1(n) = [x(n), x(n-1), \dots, x(n-N+1)]$$

$$\text{ii) } \mathbf{x}_2(n) = [y(n-1), y(n-2), \dots, x(n-N)]$$

$$\text{iii) } \mathbf{x}_3(n) = [x(n)y(n-1), \dots, x(n)y(n-N), x(n-1)y(n-1), \dots, x(n-1)y(n-N), \dots, x(n-N)y(n-N)]$$

Similar to the filter coefficient vector three signal vectors are combined to form a generalized signal vector

$$\mathbf{x}(n) = [\mathbf{x}_1(n), \mathbf{x}_2(n), \mathbf{x}_3(n)]$$

Therefore the output signal from the bilinear filter can be written as

$$y(n) = \mathbf{w}(n) \mathbf{x}^T(n)$$

The error signal of the ANC is obtained as

$$e(n) = d(n) - y(n) * b(n) \quad (2.33)$$

Similar to the adaptive FIR filter, the goal of the adaptive algorithm for the bilinear filter is to minimize the instantaneous square error using the steepest descent algorithm. So following the same procedure as FIR filter, the update algorithm for the adaptive bilinear filter for ANC can be obtained as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}'(n) e(n) \quad (2.34)$$

From the definition of $\mathbf{w}(n)$, equation (2.34) can be partitioned as following three independent vector equations

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \mu_1 \mathbf{x}'_1(n) e(n) \quad (2.35)$$

$$\mathbf{w}_2(n+1) = \mathbf{w}_2(n) + \mu_2 \mathbf{x}'_2(n) e(n) \quad (2.36)$$

$$\mathbf{w}_{i,j}(n+1) = \mathbf{w}_{i,j}(n) + \mu_3 \mathbf{x}'_3(n) e(n) \quad (2.37)$$

The simulation results provided by Sen M. Kuo and Hsien-Tsai Wu [37],[38] proved that bilinear filters not only have better MSE performance but also faster convergence speed compare to FIR and Volterra filters under strong nonlinearity situation. But because of feedback the stability of the bilinear filter is always doubtful.

2.10.3 Radial Basis Function (RBF) Approach

Riyanto T. Bambang, Lazuardi Anggono and Kenko Uchida [39] reported an ANC using radial basis function neural network. As shown in the fig. 2.18 and fig. 2.19, it consist of two stages, first, the nonlinear secondary path is identified using a radial basis function network and its learning algorithm. Secondly another radial basis function

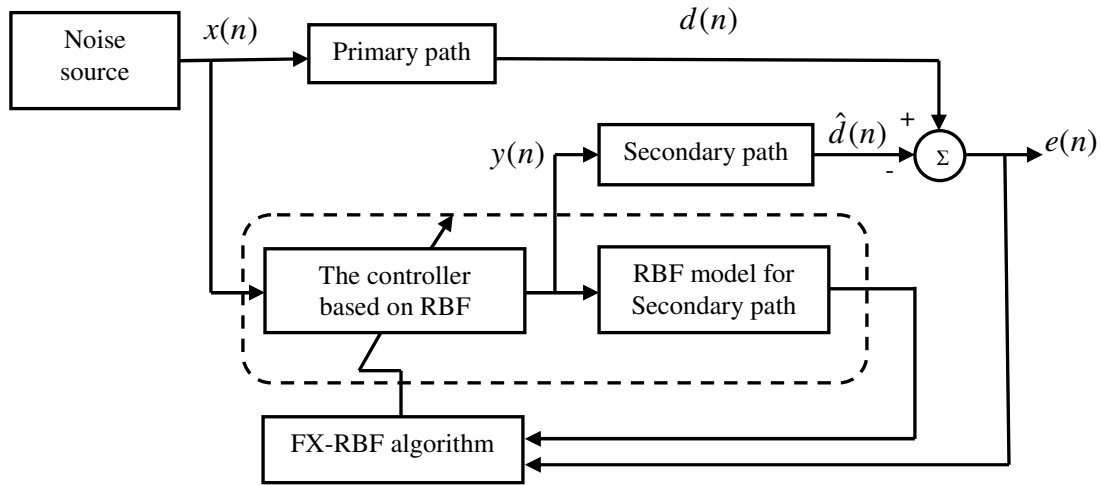


Fig. 2.18 Block diagram for ANC using RBF.

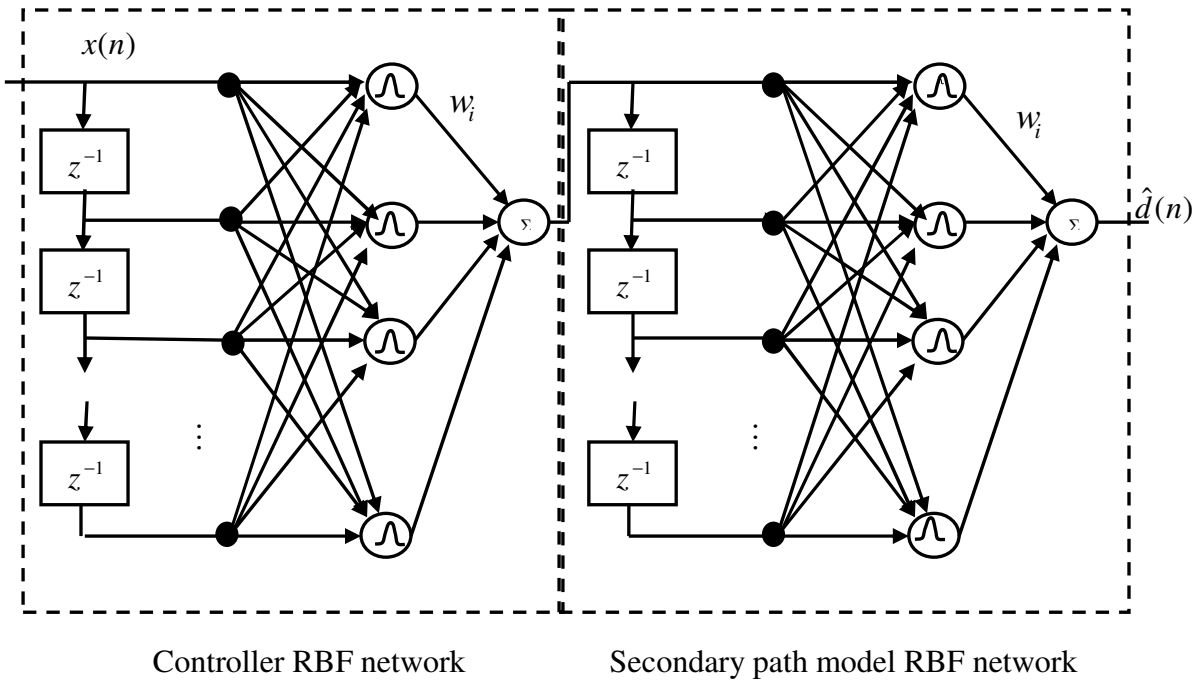


Fig. 2.19 Structure of the two RBF networks for ANC.

network is used for the controller which generates the antinoise. The algorithm developed turned out to be filtered-x version of RBF algorithm and so named FX-RBF algorithm. Real time implementation was conducted to evaluate the proposed ANC system. The centre of Gaussian functions for both model and controller network were trained using K-means clustering algorithm. Riyanto T. Bambang [40], [41] designed an ANC using recurrent radial basis function network where the strategy was analogous to the previous described work but instead of RBF network recurrent RBF network was used. The recurrent RBF network used is shown in fig. 2.20. The delayed and weighted output of each hidden node is feedback to the same node.

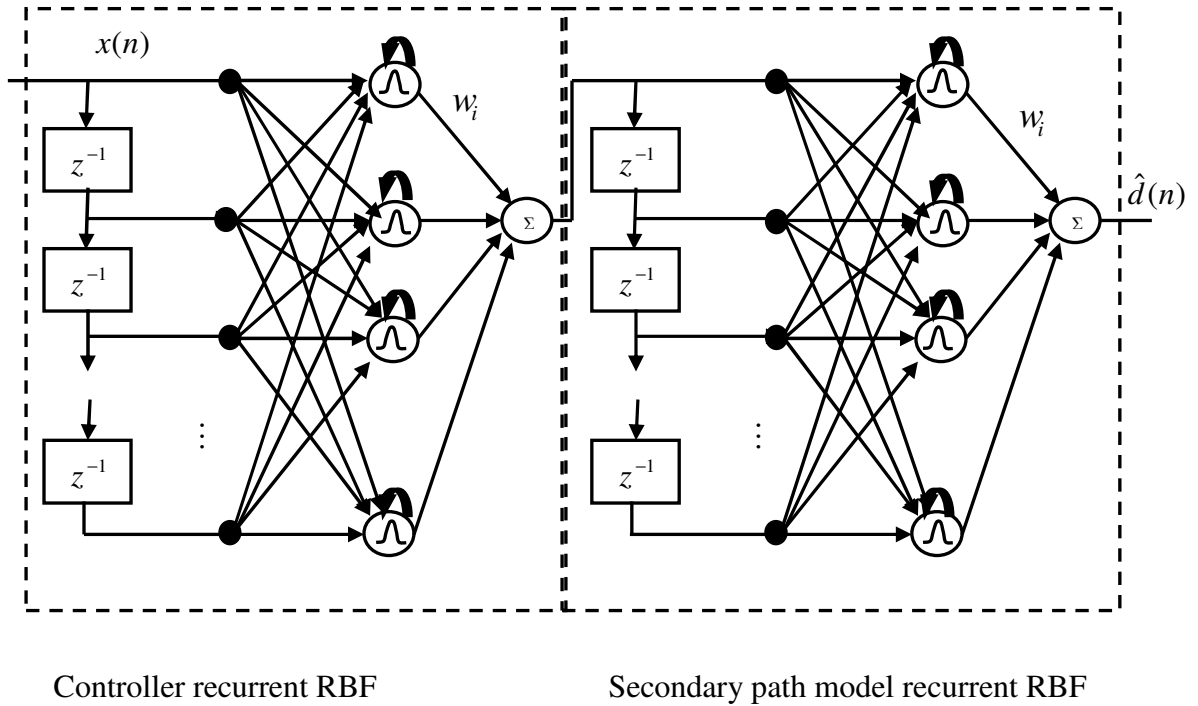


Fig. 2.20 Structure of the two recurrent RBF networks for ANC.

Jiang Lifei [42] developed an ANC based on the analysis of the characteristics of practical noises in a driver's cab, using radial basis function neural network. The training algorithm was also provided in the paper with results. Tokhi and Wood [55] also used RBF for nonlinear ANC.

2.10.4 Fuzzy Logic (FL) Approach

Since the introduction of fuzzy sets by Zadeh in 1965, many researchers have applied this theory to diverse engineering topics. In general, the knowledge base and rule

base are designed from expert experience, and then an inference engine is selected to make up a fuzzy system. C. Y. Chang and K. K. Shyu [43] reported a simple architecture to apply fuzzy logic to nonlinear ANC. They proposed seven rules to form a fuzzy FIR filter, which acts as an antinoise filter to cancel undesired noise. The developed algorithm resembles FXLMS algorithm so it is called fuzzy filtered-x algorithm. This method tunes the free parameters automatically and changes the IF–THEN rules adaptively to minimize the residual noise as new information becomes available.

Traditionally, active noise controllers are designed on the basis of a mathematical description of the plant and its linearized model. However, it is difficult to control undesired noise in a nonlinear duct plant. The conventional filtered-x based ANC systems often require hundreds of weights to control undesired noise. Therefore, resulting numerical errors, such as the round-off and quantization error are inevitable. Therefore self-tuned fuzzy-based ANC system can process both the numerical data and linguistic information to adapt the ANC system. Another advantage of the proposed algorithm includes the reduction of system complexity and the property of nonlinear compensation.

Cheng-Yuan Chang and Kuo-Kai Shyu [44] proposed a self-tuning fuzzy filtered-u algorithm which instead of complex designing procedures of traditional algorithms uses few mathematical transfer functions to design the ANC system. They have also provided a fuzzy-based self-tuning algorithm is to adjust the free parameters of the fuzzy filtered-u algorithm. In addition, the proposed method protects ANC systems against unstable poles which occur in conventional filtered-u design.

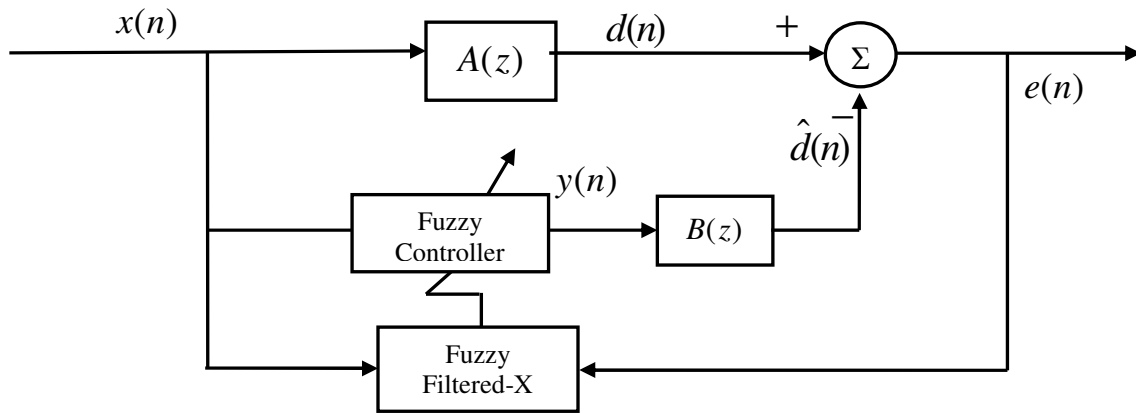


Fig. 2.21 Block diagram of ANC using fuzzy logic.

2.10.5 Fuzzy-Neural Approach

Huynh Van Tuan and Duong Hoai Nghia [45] reported a fuzzy-neural network for feedback active noise controller. The fuzzy –neural model they have used is shown in the fig. 2.22 where the term node, G, represent a Gaussian membership function to express the input fuzzy linguistic variables and “rule nodes, R” represent the fuzzy rules. Node, N, performs the normalization of the firing strengths coming from pervious layer. They also developed fuzzy neural-based filtered-x least-mean-square algorithm and proved the convergence of the algorithm using a discrete Lyapunov function.

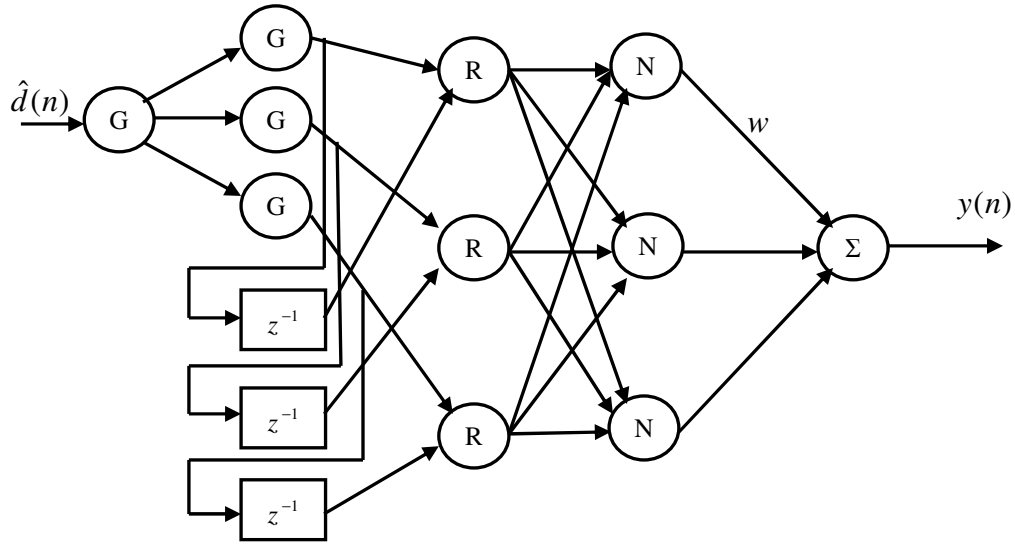


Fig. 2.22 Structure of neuro-fuzzy controller used for ANC.

Jian Liu, Jinwei Sun and Guo Wei [46] proposed a new narrowband ANC system using an ANFIS (adaptive neuro-fuzzy inference system) as an adaptive controller. For the purpose of computational cost reduction, the nonlinear premise parameters in the ANFIS are kept fixed and only its linear consequent parameters are adjusted based on a gradient descent method. The block diagram is shown in fig. 2.23.

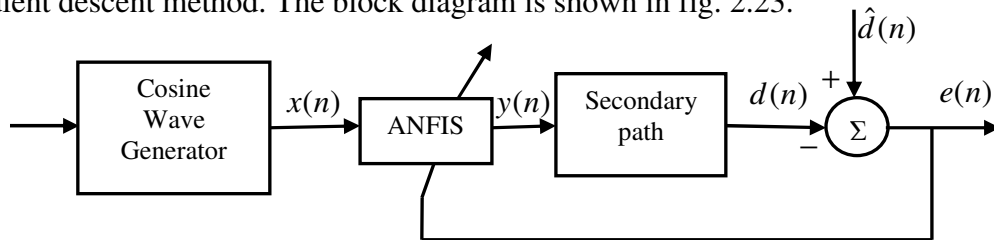


Fig. 2.23 Block diagram of ANC using ANFIS.

Navid Azadi, Abdolreza Ohadi [47] proposed an enhanced multi-channel active fuzzy neural network noise controller in a rectangular enclosure. The block diagram of the ANC system they used and the fuzzy neural network they employed is shown in the fig.2.24 and fig. 2.25 respectively. They proposed a multi-channel enhanced fuzzy neural network (EFNN) error back propagation algorithm while taking into account the secondary path effect in derivation of equations. The results provided shown that enhanced FNN algorithm outperformed FXLMS algorithm when there is a highly nonlinear primary path in the ANC system.

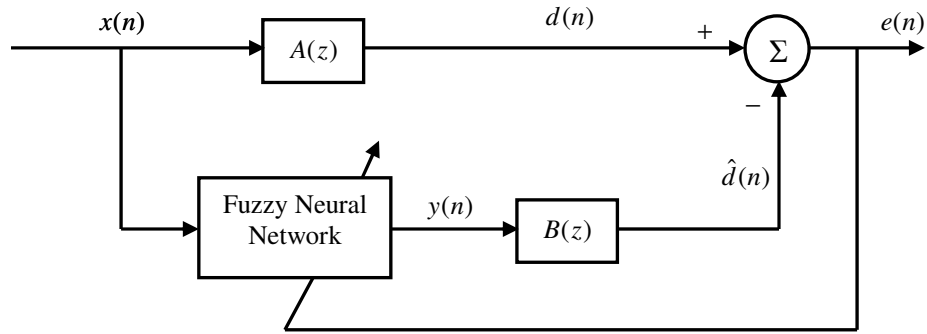


Fig. 2.24 Block diagram of ANC using fuzzy-neural network.

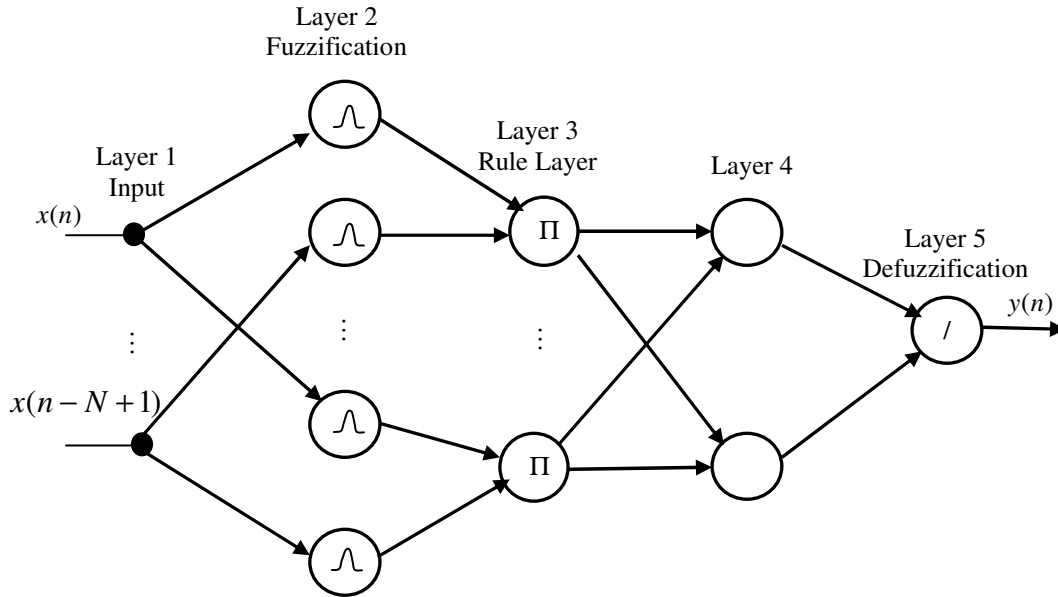


Fig. 2.25 Structure of fuzzy-neural network controller for ANC.

2.10.6 Genetic Algorithm (GA) Approach

Fabrizio Russo and Giovanni L. Sicuranza [48], [49] investigated the performance of genetic optimization for nonlinear active noise control based on nonlinear Volterra filters. They showed two advantages of using genetic algorithm (GA) for nonlinear ANC problem. i) While standard filtered-x algorithms may converge to local minima, GA may handle this problem efficiently. ii) This class of algorithms does not require the pre-identification of the secondary paths because unlike the class of FXLMS algorithm, estimated secondary path transfer function is not required in adaptation process. Here GA is used to optimize the filter coefficients of the nonlinear Volterra filter. The block diagram they suggested is reproduced in the fig. 2.26 below.

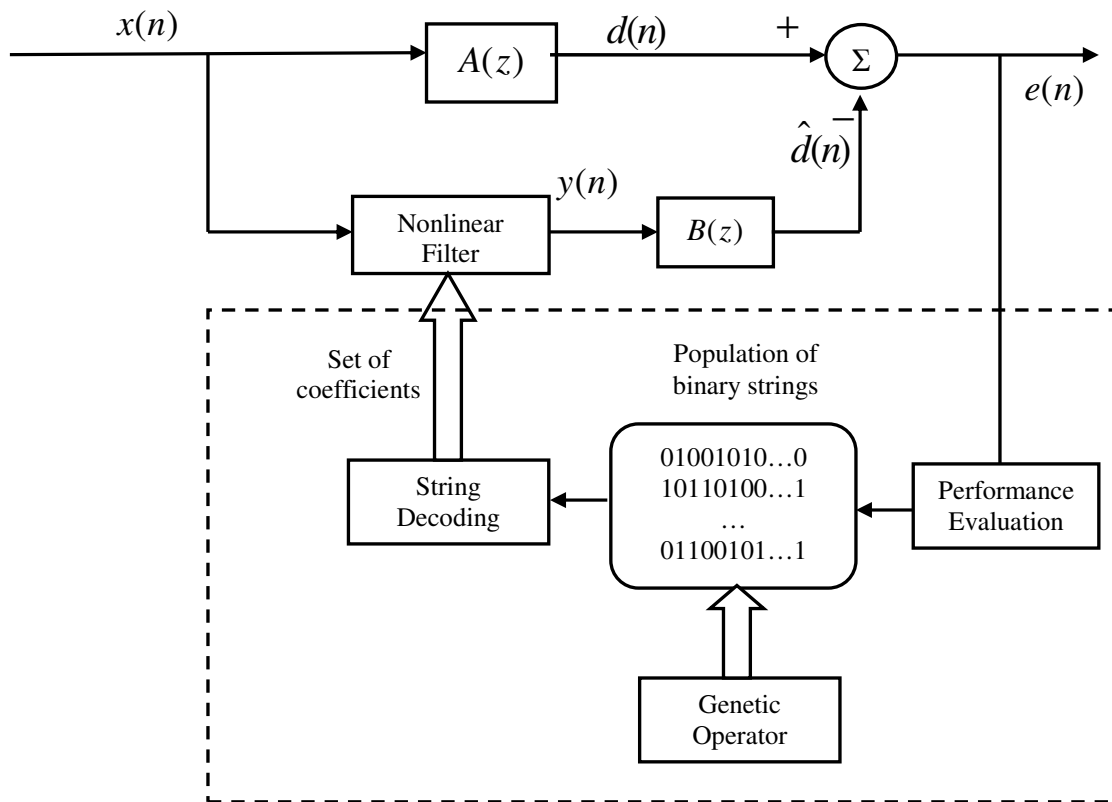


Fig.2.26 Block diagram of the genetic ANC system. (Fabrizio Russo and Giovanni L. Sicuranza [48]).

Cheng-Yuan Chang and Deng-Rui Chen [50] proposed an adaptive genetic algorithm (AGA) for an ANC system to eradicate the problem of FXLMS algorithm converging to local minimum. The conventional ANC system implements the FXLMS algorithm to update the coefficients of the linear finite-impulse response (FIR) filters. For nonlinear ANC using nonlinear Volterra filters require Volterra FXLMS (VFXLMS) algorithm. The proposed method replace FXLMS algorithm with FIR adaptive genetic algorithm (FAGA) for FIR filters and VFXLMS algorithm with Volterra adaptive genetic algorithm (VAGA) for Volterra filters. The proposed AGA method does not require pre-identifying the secondary path for the ANC operation thereby making the system immune to secondary path identification error. Performance of the FAGA and VAGA are compared with FXLMS and VFXLMS algorithms.

2.10.7 Particle Swarm Optimization (PSO) Approach

Nirmal Kumar Rout, Debi Prasad Das and Ganapati Panda [51] presented a new online ANC algorithm using PSO-based training. Particle swarm optimization (PSO) is a nongradient but simple evolutionary computing-type algorithm. Conventionally PSO-based algorithm is shown to be ineffective to regain convergence in the case of occurrence of an abrupt change in primary and/or secondary paths (time varying primary path and/or secondary path). To cope with this problem, in the paper, the conventional PSO algorithm is modified to introduce a new conditional reinitialized particle swarm optimization (CRPSO) algorithm which is used to optimize the weights of an FIR filter. The added advantage of the algorithm is that it doesn't require pre-estimation of secondary path, thereby, making the system immune to secondary path identification error. Performance of the PSO is compared with FXLMS algorithm and GA based algorithms. The strategy adopted by Nirmal Kumar Rout, Debi Prasad Das and Ganapati Panda [51] is represented by the block diagram below, fig 2.27. Nithin V. George and Ganapati Panda presented a robust evolutionary active noise control system using Wilcoxon norm and particle swarm optimization algorithm [80].

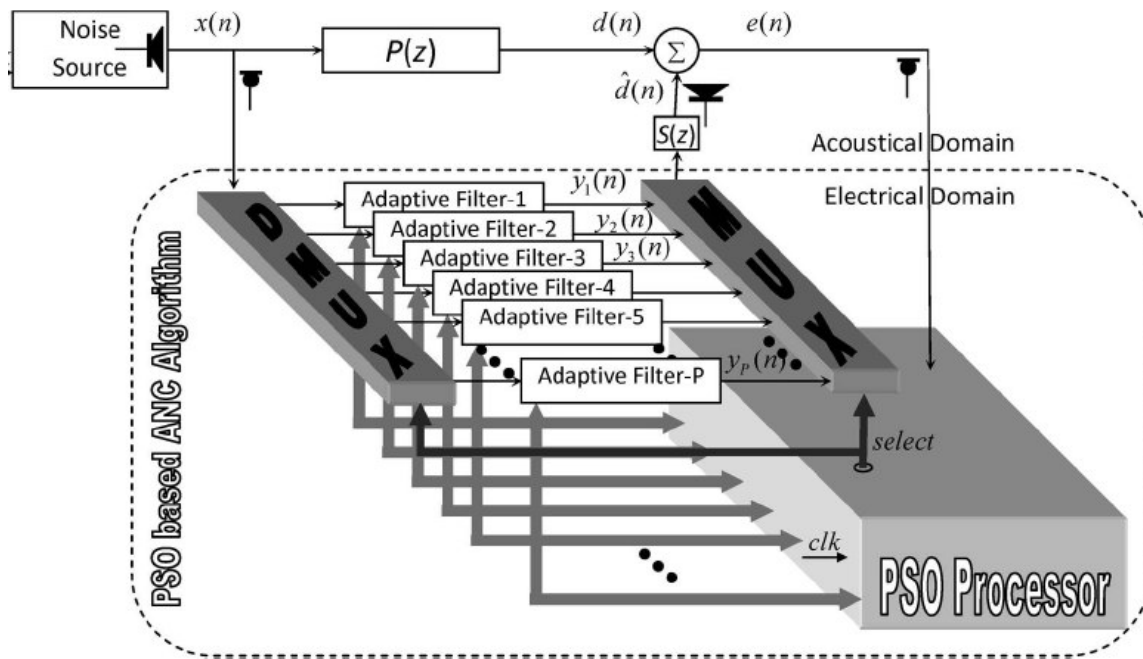


Fig. 2.27 Block diagram of PSO-based training of an ANC system (Nirmal Kumar Rout, Debi Prasad Das and Ganapati Panda [51]).

2.11 Summary

A comprehensive study of existing ANC schemes have been carried out in this chapter. The study categorize the available ANC schemes into two categories, i) study of linear ANC ii) study of nonlinear ANC. Literature is rich and well documented for linear ANC where as research is continuing for nonlinear ANC. Various sources, responsible for introducing nonlinearity into the ANC system are highlighted. Strategies adopted for applications of soft computing techniques like multi-layer artificial neural network, radial basis function network, recurrent radial basis function network, fuzzy logic, neuro-fuzzy technique, etc. and evolutionary techniques like genetic algorithm and particle swarm optimization, etc. for nonlinear ANC have been presented. The necessity of ANC in diverse field of engineering and technology are discussed. The role played by ANC and the apparent benefits of employing an ANC in these applications are analyzed extensively. The problems encountered in the real time applications of ANC are also highlighted. Some of the broad conclusions made from the study are as follows

- Extensive documentation exists with reference to linear ANC, so analysis can be done easily.
- The very introduction of nonlinearity even in a carefully designed ANC degrades the quality of noise cancellation quite significantly and hence must be taken into consideration to improve the ANC performance.
- More indepth investigation is needed to develop improved techniques for nonlinear ANC to yield better noise control performance.
- Research can be focused on one of the following subcategories
 - Development of new architecture or structure for the ANC controller.
 - Development of new algorithms which will support the new structures.
 - Modifications in the existing schemes of ANC with specific emphasis to structural minimization without sacrificing performance.
 - Development of new strategy to reduce computational complexity requirement from the implementation point of view.
 - Customizing the developed schemes for different application areas.
 - Utilize the recent knowledge of DSP, soft computing and evolutionary techniques for achieving better performance.

CHAPTER 3

Neural Network Approach to Nonlinear Active Noise Control

3.1 Background

Active noise control (ANC) has attracted a lot of research interest because of rapid increase of acoustical noise pollution and insufficiency of passive techniques for noise control. In an ANC, noise is deliberately introduced with an objective to cancel another undesirable noise. ANC employ the superposition principle, where the undesired noise is reduced by adding another noise called antinoise with the same amplitude but opposite polarity. Antinoise is generated by actuators such as loudspeaker. Linear adaptive FIR filter along with the filtered-x LMS (FXLMS) algorithm is the most common strategy applied in both feed-forward and feedback ANC due to its ease in implementation [2]. It has been mentioned earlier (detail discussion in this chapter) that there are a number of sources of nonlinearity in an ANC. This gives rise to a situation where linear FIR filter with FXLMS algorithm will show performance degradation and even fail in some situations. This is due to the fact that a linear type adaptive system has been used for approximating a system which exhibit nonlinear characteristics. Thus avenues are wide open to use adaptive systems with nonlinear approximation capability like artificial neural network, fuzzy logic and polynomial filter etc. This chapter is entirely devoted to the study of artificial neural network for nonlinear ANC.

3.2 Multilayer Artificial Neural Network (MLANN)

It has already been established that multilayer artificial neural network based ANCs have significant performance improvement over the conventional linear ANCs based on FXLMS or FXRLS adaptive algorithms [52]-[61]. The basic objective of the present work is primarily aimed at developing new ANC in the MLANN domain having reduced structural complexity so that these can be easily implemented in real-time. The

learning algorithms developed to update the weights of different structures utilize the backpropagation algorithm concept. However, with reference to presence of secondary path certain modifications have been included in the weight adaptation equation.

In [52] Snyder and Tanaka proposed feedforward control of vibration using a neural network-based control system, with the aim to derive an architecture which might be capable of supplanting the commonly used FIR filter with FXLMS algorithm. Fig.3.1 shows the ANC developed by Snyder and Tanaka. They have employed two multilayer perceptron (MLP) networks, one network was exclusively employed to model the nonlinear secondary path. This network is trained offline using standard backpropagation algorithm while the ANC is not in operation. Once converged its weights are frozen and are, latter on, used for stable operation of the ANC. The other network is called the controller network which is used to produce the antinoise. At the first glance this problem may seem trivial, as application of the standard backpropagation algorithm can update the weights. But owing to the presence of the tapped delay line input to the secondary path model, the standard backpropagation algorithm cannot be used directly in this arrangement as it must backpropagate through a tapped delay line. Therefore, the standard backpropagation algorithm is modified resulting in a formulation of a new algorithm which enables stable adaptation of the neural controller. The algorithm was shown to be simply a generalization of the linear filtered-x LMS algorithm.

Martin Bouchard, Bruno Paillard, and Chon Tan Le Dinh [53] presented an improved training algorithm for the neural network in ANC. They used the same neural network structure as Snyder and Tanaka [52] but introduced new heuristical training algorithms with the objective to develop faster convergence speed (by using nonlinear recursive-least squares algorithms) and/or lower computational loads (by using an alternative approach to compute the instantaneous gradient of the cost function). The block diagram of ANC using neural network is shown in fig. 3.1

In [54] Martin Bouchard introduced a heuristic procedure for the development of recursive-least-squares algorithms based on the filtered-x and the adjoint gradient approaches. He also used the same structure for ANC, as in [52] and [53], employing two multilayer feedforward neural networks. But he developed new recursive-least-squares algorithms for the training of the neural network controller in the two network ca-

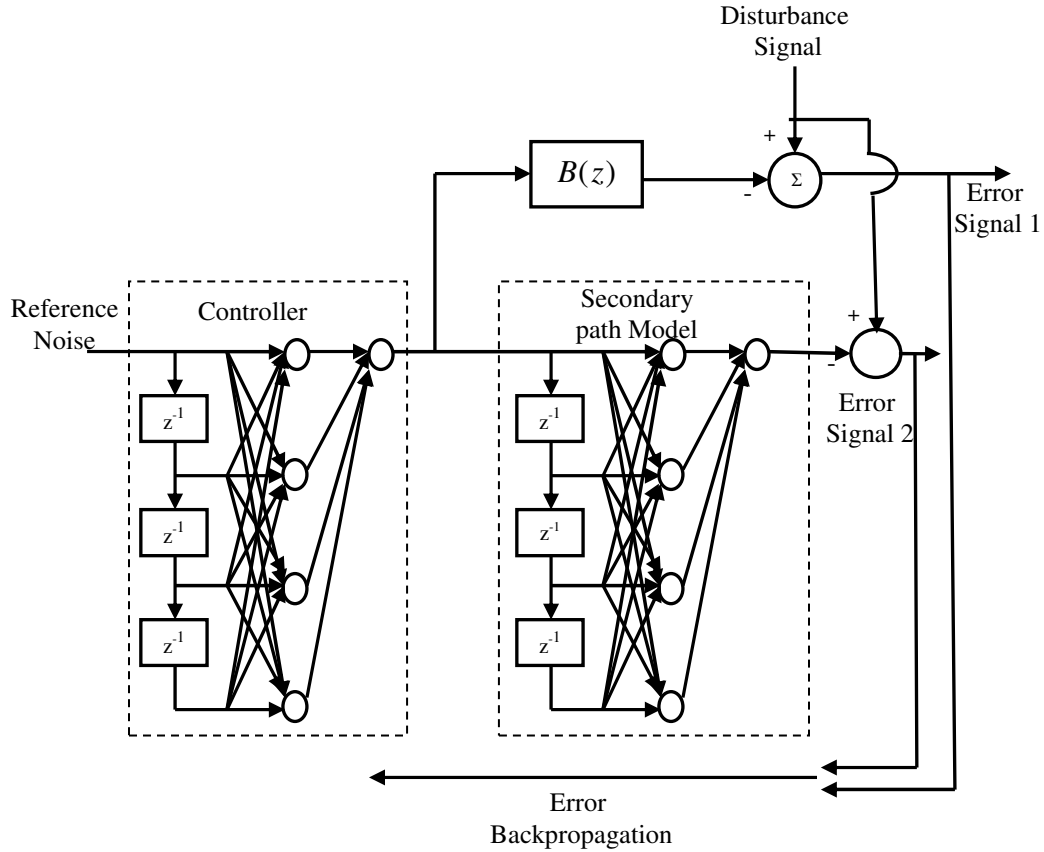


Fig. 3.1 Block diagram of ANC using two neural networks.

scaded structure. It has been seen from the results that these new algorithms yielded better convergence performance than previously published algorithms.

Cheng-Yuan Chang and Fang-Bor Luoh [56] and Cheng-Yuan Chang and Shing-Tai Pan [60] presented a new architecture for nonlinear ANC using only one multilayer feedforward network. The ANC they developed is shown in fig.3.2. Here the neural network is used for ANC controller where as the secondary path is modeled using a linear adaptive FIR filters. The network has only three layers, input layer, hidden layer and output layer. Output layer has only one node where as other layers have more than one nodes. The activation function of both the hidden layer nodes and output layer node are taken as linear function, basically to avoid premature saturation of back propagation algorithm. They developed a new neural network update algorithm using

backpropagation and gradient descent methods. The ANC architecture and the developed algorithm are simple and easy to implement but the major drawback of this technique is it employ a FIR filter for secondary path modeling. As discussed earlier, in actual implementation of ANC there exist a number of factors which contribute to nonlinearity of secondary path. So using adaptive FIR filter for secondary path modeling degrades the performance of the ANC.

3.3 Proposed Neural Network Technique

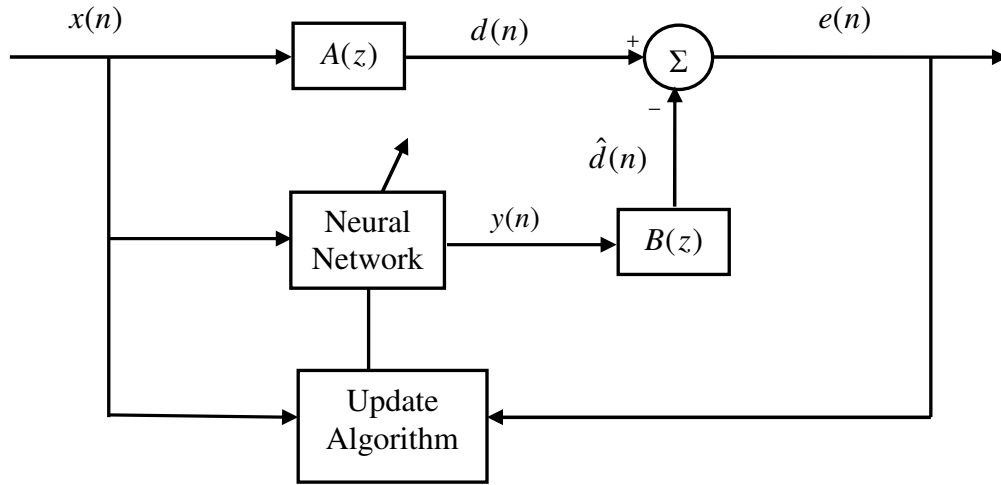


Fig 3.2 Block diagram of ANC using neural network.

In this chapter, a feedforward nonlinear active noise control system employing a multilayer neural network is developed. The multilayer neural network involved is shown in the fig.3.3. Two separate update algorithms are derived for two situations. First algorithm is derived when secondary path is modeled as a FIR filter (which assume secondary path to be linear). Another algorithm is developed when secondary path is modeled as difference equation representing the nonlinear secondary path. Both the algorithms are found to be extended version of filtered-x LMS algorithm.

Filtered-e LMS is an algorithm for ANC well known for its low computational complexity requirement. Both the developed algorithms are modified to accommodate filtered-e LMS algorithm which results in reduced computational complexity. Performances of the proposed algorithms are validated through extensive computer simulations.

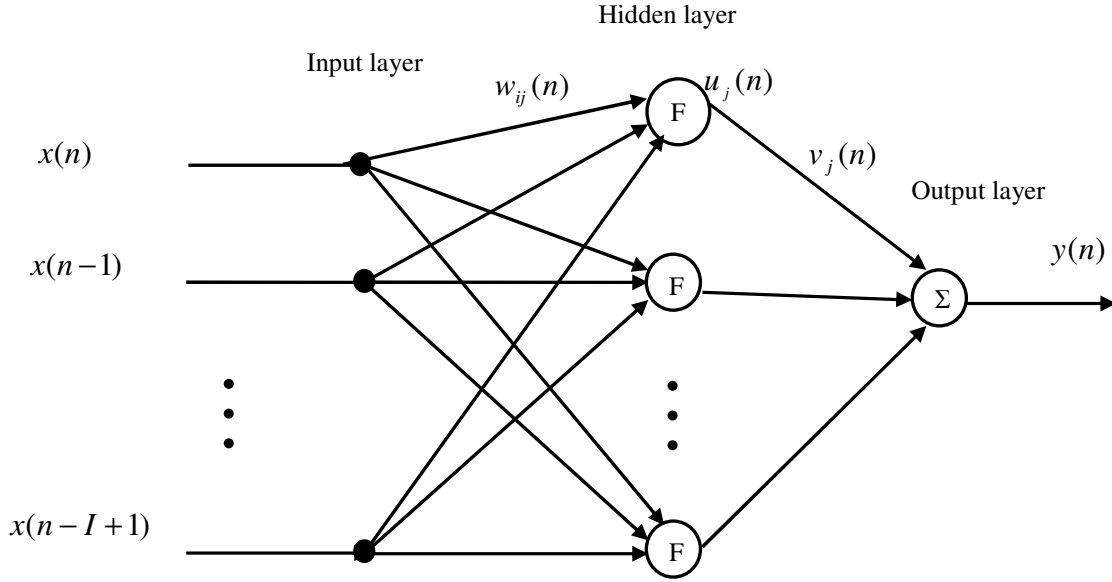


Fig 3.3 Neural network controller.

The neural network which will be used as the controller for NANC is shown in fig. 3.3. The network has three layers, input layer, hidden layer and output layer. Output layer has only one neuron but hidden layer may have many neurons. Neurons in the hidden layer have nonlinear activation function where as the neuron in the output layer has linear activation function. Total number of neurons of input layer and neurons in the hidden layer are I and J respectively. The network has IJ numbers of hidden layer synaptic weights represented by $w_{ij}(n)$ where $i=0,1,2,3,\dots,I-1$ and $j=0,1,2,3,\dots,J-1$. The network has J number of output layer synaptic weights v_j , $j=0,1,2, \dots J-1$.

The net internal activity level $c_j(n)$ for j^{th} hidden layer neuron at n^{th} instant is

$$c_j(n) = \sum_{i=0}^{I-1} x(n-i)w_{ij}(n) \quad \text{for } j=0,1, 2, \dots ,J-1 \quad (3.1)$$

Considering F as the activation function, the output of the j^{th} hidden layer neuron at n^{th} instant is calculated as

$$u_j(n) = F(c_j(n)) \quad (3.2)$$

The final output from the network, at n^{th} instant, is computed considering the node in the output layer to be a summing unit as given by

$$y(n) = \sum_{j=0}^{J-1} v_j(n) u_j(n) \quad (3.3)$$

The final output of the neural network is passed through the secondary path to generate the antinoise, $\hat{d}(n)$. The equation for the antinoise generated by the ANC is given by

$$\hat{d}(n) = \sum_{k=0}^{M-1} b_k(n) y(n-k) \quad (3.4)$$

where b_k is k^{th} coefficient of impulse response of secondary path model and M is the order of secondary path filter. For the ANC, $d(n)$ is the undesired noise to be cancelled at the zone of silence. After the noise cancellation process is over the residual noise is sensed by the error microphone. This residual noise called error signal, at n^{th} instant, is calculated as follows

$$e(n) = \hat{d}(n) - d(n) \quad (3.5)$$

The error signal $e(n)$ actuates a control mechanism, the purpose of which is to apply corrective adjustments to the synaptic weights. This objective is achieved by minimizing a cost function or index of performance, $\xi(n)$. A commonly used cost function based on the mean-squared-error criterion has been applied here.

$$\begin{aligned} \xi(n) &= \frac{1}{2} [e^2(n)] \\ &= \frac{1}{2} [d(n) - \hat{d}(n)]^2 \\ &= \frac{1}{2} \left[d(n) - \sum_{k=0}^{M-1} b_k(n) u(n-k) \right]^2 \end{aligned} \quad (3.6)$$

Now the synaptic weight update equation can be derived. The update equations are formulated separately for two situations

- secondary path is assumed linear (modeled by FIR filter)
- secondary path is assumed nonlinear (modeled by a difference equation having higher order terms and cross terms)

3.3.1 Linear Secondary Path

When the secondary path is linearly modeled, the synaptic weight update equations for output layer and hidden layer are derived separately.

Output layer synaptic weight update

Using back-propagation and gradient descent methods the synaptic weights of output layer are updated by adding a negative gradient of the cost function with respect to the weights of interest

$$v_j(n+1) = v_j(n) - \mu \Delta v_j(n) \quad j=0, 1, 2, \dots, J-1 \quad (3.7)$$

where μ is the learning rate (step size)

Above equation can also be written as

$$\mathbf{v}(n+1) = \mathbf{v}(n) - \mu \Delta \mathbf{v}(n) \quad (3.8)$$

where $\mathbf{v}(n) = [v_0(n), v_1(n), \dots, v_{J-1}(n)]$ is the output layer synaptic weight vector.

The gradients of cost function with respect to output layer synaptic weights is defined as follows

$$\Delta v_j(n) = \frac{\partial \xi(n)}{\partial v_j(n)} \quad (3.9)$$

Putting (3.6) in the above equation we get

$$\begin{aligned} \Delta v_j(n) &= \frac{\partial \xi(n)}{\partial v_j(n)} \\ &= \frac{1}{2} \frac{\partial [e^2(n)]}{\partial v_j(n)} \\ &= e(n) \frac{\partial [d(n) - \hat{d}(n)]}{\partial v_j(n)} \end{aligned}$$

Since undesired noise $d(n)$ is independent of the neural network synaptic weights, the following relation is obtained

$$\frac{\partial \xi(n)}{\partial v_j(n)} = -e(n) \frac{\partial \hat{d}(n)}{\partial y(n)} \frac{\partial y(n)}{\partial v_j(n)} \quad (3.10)$$

Taking the expressions for $\hat{d}(n)$ and $y(n)$ from (3.4) and (3.5) respectively and putting in the above equation it is found that

$$\frac{\partial \xi(n)}{\partial v_j(n)} = -e(n) \sum_{k=0}^{M-1} b_k(n) u_j(n-k) \quad (3.11)$$

where M is the order of secondary path model and $b_k(n)$ is the k^{th} coefficient of secondary path model impulse response. Putting (3.11) in the (3.7), update equation for output layer synaptic weights are as follows

$$v_j(n+1) = v_j(n) + \mu e(n) \sum_{k=0}^{M-1} b_k(n) u_j(n-k) \quad (3.12)$$

Hidden layer synaptic weight update

Similar to output layer synaptic weights, the hidden layer synaptic weights are updated as

$$w_{ij}(n+1) = w_{ij}(n) - \mu \Delta w_{ij}(n) \quad (3.13)$$

where $i=0, 1, \dots, I-1$ and $j=0, 1, \dots, J-1$

The update equation for hidden layer weights can also be written as follows

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \Delta \mathbf{w}(n) \quad (3.14)$$

where $\mathbf{w}(n)$ the hidden layer synaptic weight matrix, at n^{th} instant, defined by

$$\mathbf{w}(n) = \begin{bmatrix} w_{00} & w_{01} & \cdots & w_{0J-1} \\ w_{10} & w_{11} & \cdots & w_{1J-1} \\ \vdots & \vdots & \cdots & \vdots \\ w_{I-10} & w_{I-11} & \cdots & w_{I-1J-1} \end{bmatrix}$$

The gradient of the cost function with respect to hidden layer synaptic weights is

$$\begin{aligned} \Delta w_{ij}(n) &= \frac{\partial \xi(n)}{\partial w_{ij}(n)} \\ &= \frac{1}{2} \frac{\partial [e^2(n)]}{\partial w_{ij}(n)} \\ &= e(n) \frac{\partial [d(n) - \hat{d}(n)]}{\partial w_{ij}(n)} \end{aligned}$$

Since the desired signal is independent of the hidden layer synaptic weights

$$\begin{aligned} \Delta w_{ij}(n) &= -e(n) \frac{\partial [\hat{d}(n)]}{\partial w_{ij}(n)} \\ &= -e(n) \sum_{k=0}^{M-1} \frac{\partial \hat{d}(n)}{\partial y(n-k)} \frac{\partial y(n-k)}{\partial w_{ij}(n)} \end{aligned}$$

$$= -e(n) \sum_{k=0}^{M-1} b_k(n) \frac{\partial y(n-k)}{\partial w_{ij}(n)} \quad (3.15)$$

Assuming that synaptic weights are adapted slowly it can be written that

$$\begin{aligned} \frac{\partial y(n-k)}{\partial w_{ij}(n)} &\approx \frac{\partial y(n-k)}{\partial w_{ij}(n-k)} \\ &= \frac{\partial y(n-k)}{\partial u_j(n-k)} \frac{\partial u_j(n-k)}{\partial w_{ij}(n-k)} \end{aligned}$$

Taking the expressions for $y(n-k)$ and $u_j(n-k)$ from (3.3) and (3.2) and putting in the above equation it is found that

$$\frac{\partial y(n-k)}{\partial w_{ij}(n)} = v_j(n-k) F'(c_j(n-k)) \frac{\partial c_j(n-k)}{\partial w_{ij}(n-k)}$$

Taking the value of $c_j(n-k)$ from (3.1) and putting in above equation it is observed that

$$\frac{\partial y(n-k)}{\partial w_{ij}(n)} = v_j(n-k) F'(c_j(n-k)) x(n-i-k) \quad (3.16)$$

Putting values of (3.16) and (3.15) in (3.13) the update equation for hidden layer synaptic weights can be expressed as given below

$$w_{ij}(n+1) = w_{ij}(n) + \mu e(n) \sum_{k=0}^{M-1} b_k(n) v_j(n-k) F'(c_j(n-k)) x(n-i-k) \quad (3.17)$$

where, $i=0, 1, 2, \dots, I-1$. and $j=0, 1, 2, \dots, J-1$

3.3.2 Nonlinear Secondary Path

The equations expressed in the previous section are valid when secondary path is modeled as linear FIR filter. To take care of the nonlinearity in the secondary path, the secondary path can be modeled as a difference equation having higher order terms and cross terms. An example is given below

$$\hat{d}(n) = y(n) + 0.8y(n-1) + \underbrace{0.6y^2(n-2)}_{\text{square term}} + \underbrace{0.4y(n-2)y(n-3) + 0.2y(n-3)y(n-5)}_{\text{cross terms}}$$

where $y(n)$ and $\hat{d}(n)$ are input and output of the secondary path filter model.

For this type of nonlinear secondary path model the synaptic weight update equation derived in the previous section have to be modified to develop a complete set of new equations.

Output layer synaptic weight update

The synaptic weights of output layer are updated by adding a negative gradient of the cost function with respect to the weights of interest

$$v_j(n+1) = v_j(n) - \mu \Delta v_j(n) \quad j=0, 1, 2, \dots, J-1 \quad (3.18)$$

where μ is the learning rate (step size) and

$$\Delta v_j(n) = \frac{\partial \xi(n)}{\partial v_j(n)}$$

The gradient of the cost function with respect to output layer synaptic weights is

$$\begin{aligned} \Delta v_j(n) &= \frac{\partial \xi(n)}{\partial v_j(n)} \\ &= \frac{1}{2} \frac{\partial [e^2(n)]}{\partial v_j(n)} \end{aligned}$$

Putting the expression of $e(n)$ in (3.5) it is found that

$$\frac{\partial \xi(n)}{\partial v_j(n)} = e(n) \frac{\partial [d(n) - \hat{d}(n)]}{\partial v_j(n)}$$

Since the undesirable noise $d(n)$ is independent of neural network synaptic weights

$$\begin{aligned} \frac{\partial \xi(n)}{\partial v_j(n)} &= -e(n) \frac{\partial \hat{d}(n)}{\partial v_j(n)} \\ &= -e(n) \sum_{k=0}^{M-1} \frac{\partial \hat{d}(n)}{\partial y(n-k)} \frac{\partial y(n-k)}{\partial v_j(n)} \end{aligned} \quad (3.19)$$

Assuming that the weights are adapted slowly the second term of the summation can be written as

$$\begin{aligned} \frac{\partial y(n-k)}{\partial v_j(n)} &\approx \frac{\partial y(n-k)}{\partial v_j(n-k)} \\ &= u_j(n-k) \end{aligned}$$

Putting this value in equation (3.19) it is found that

$$\frac{\partial \xi(n)}{\partial v_j(n)} = -e(n) \sum_{k=0}^{M-1} \frac{\partial \hat{d}(n)}{\partial y(n-k)} u_j(n-k) \quad (3.20)$$

The first term of the summation is represented as follows

$$\begin{aligned} \frac{\partial \hat{d}(n)}{\partial y(n-k)} &= \left[\frac{\partial \hat{d}(n)}{\partial y(n)}, \frac{\partial \hat{d}(n)}{\partial y(n-1)}, \frac{\partial \hat{d}(n)}{\partial y(n-2)}, \dots, \frac{\partial \hat{d}(n)}{\partial y(n-M+1)} \right] \\ &= [\tilde{b}_0(n), \tilde{b}_1(n), \tilde{b}_2(n), \dots, \tilde{b}_{M-1}(n)] \\ &= \tilde{\mathbf{b}}(n) \quad k=0, 1, 2, \dots, M-1 \end{aligned} \quad (3.21)$$

Here a FIR filter can be defined with $\tilde{b}_0(n), \tilde{b}_1(n), \tilde{b}_2(n), \dots, \tilde{b}_{M-1}(n)$ as its coefficients.

But here $\tilde{b}_0(n), \dots, \tilde{b}_{M-1}(n)$ are varying with respect to time. But still a filter with time varying filter coefficients can be defined as

$$\tilde{\mathbf{b}}(n) = [\tilde{b}_0(n), \tilde{b}_1(n), \tilde{b}_2(n), \dots, \tilde{b}_{M-1}(n)]^T \quad (3.22)$$

This filter, represented as $\tilde{\mathbf{b}}(n)$, is called virtual secondary path filter [62].

Putting (3.21) and (3.22) in (3.20) the following is obtained

$$\frac{\partial \xi(n)}{\partial v_j(n)} = -e(n) \sum_{k=0}^{M-1} \tilde{b}_k(n) u_j(n-k) \quad (3.23)$$

Finally putting (3.23) in weight update equation (3.18) results in the following

$$v_j(n+1) = v_j(n) + \mu e(n) \sum_{k=0}^{M-1} \tilde{b}_k(n) u_j(n-k) \quad j=0, 1, 2, \dots, J-1 \quad (3.24)$$

Hidden layer synaptic weight update

Update equation for hidden layer synaptic weights can be written as

$$w_{ij}(n+1) = w_{ij}(n) + \mu e(n) \frac{\partial \hat{d}(n)}{\partial w_{ij}(n)} \quad (3.25)$$

Proceeding in the similar line as in the case of linear secondary path the following is obtained

$$\begin{aligned} \frac{\partial \hat{d}(n)}{\partial w_{ij}(n)} &= \sum_{k=0}^{M-1} \frac{\partial \hat{d}(n)}{\partial y(n-k)} \frac{\partial y(n-k)}{\partial w_{ij}(n)} \\ &= \sum_{k=0}^{M-1} \tilde{b}_k(n) v_j(n-k) F'(c_j(n-k)) x(n-i-k) \end{aligned} \quad (3.26)$$

Finally putting (3.26) in (3.25) it is found that update equation becomes

$$w_{ij}(n+1) = w_{ij}(n) + \mu e(n) \sum_{k=0}^{M-1} \tilde{b}_k(n) v_j(n-k) F'(c_j(n-k)) x(n-i-k) \quad (3.27)$$

This algorithm is termed as neural filtered-x LMS (NFXLMS) algorithm.

3.4 Development of Neural Filtered-e LMS Algorithm

Adjoint LMS algorithm developed by E. A. Wan [14] is a simple alternative to FXLMS algorithm which reduces computational complexity specifically for multichannel ANC. Unlike FXLMS algorithm where a secondary path model is placed in the reference signal path, here an adjoint of secondary path model can be placed in the error path as shown in the fig. 3.4. In the figure $\hat{B}_-(z)$ is adjoint of secondary path model $\hat{B}(z)$. This algorithm is called filtered-error LMS or filtered-e LMS (FELMS) algorithm.

FELMS algorithm adaptive equation is

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n-M+1) e'(n) \quad (3.28)$$

where delay (equal to the order of secondary path model) is provided to reference signal to compensate for the delay in the error path. $e'(n)$ is the filtered error signal generated by filtering the error signal by the estimated adjoint secondary path filter denoted by $\hat{B}_-(z)$. Adjoint secondary path filter is obtained by simply writing the filter coefficients in the reverse order. On the basis of FELMS algorithm the synaptic weight update algorithm equations of ANC using neural network can also be modified to obtain neural filtered-e LMS (NFELMS) algorithm.

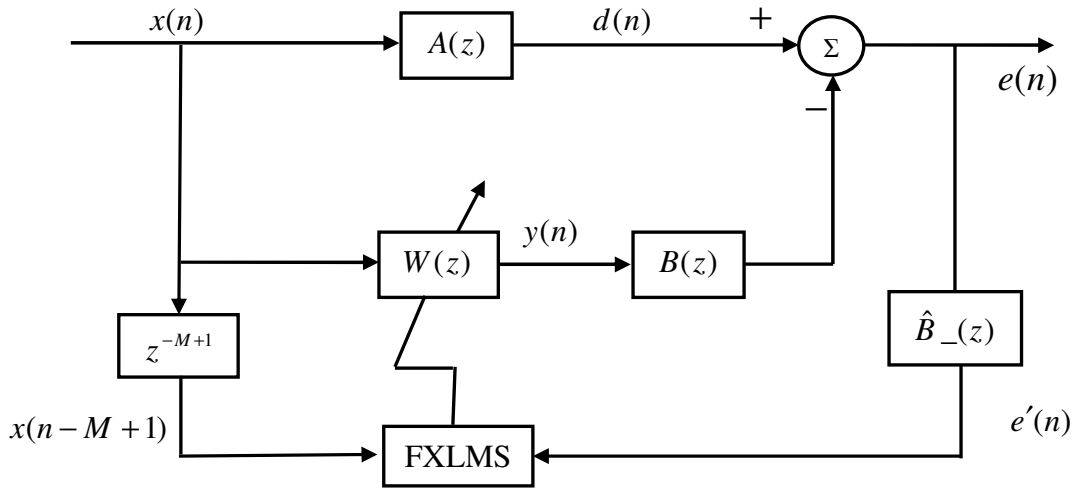


Fig 3.4 Block diagram of ANC system using FELMS algorithm.

Here again two situations are considered for discussion,

- secondary path is assumed linear (modeled by FIR filter)
- secondary path is assumed nonlinear (modeled by a difference equation having higher order terms and cross terms)

3.4.1 Linear Secondary Path

Output layer synaptic weights

The synaptic weight update equation for output layer weights is obtained by modifying (3.12) as follows.

$$\begin{aligned} v_j(n+1) &= v_j(n) + \mu u_j(n-M+1) \sum_{k=0}^{M-1} b_{M-1-k}(n) e(n-k) \\ &= v_j(n) + \mu u_j(n-M+1) e'(n) \quad j = 0, 1, 2, \dots, J-1 \end{aligned} \quad (3.29)$$

where $e'(n)$ is the error signal filtered through the estimated adjoint secondary path filter.

Hidden layer synaptic weights

Similarly the synaptic weight update equation for output layer synaptic weights is derived by modifying (3.17) as follows.

$$\begin{aligned} w_{ij}(n+1) &= w_{ij}(n) + \mu v_j(n-M+1) F(c_j(n-M+1)) \sum_{k=0}^{M-1} b_{M-1-k}(n) e(n-k) \\ &= w_{ij}(n) + \mu v_j(n-M+1) F(c_j(n-M+1)) e'(n) \end{aligned} \quad (3.30)$$

$i=0,1,\dots,I-1 \text{ and } j=0,1,\dots,J-1$

3.4.2 Nonlinear Secondary Path

Nonlinearity in secondary path is frequently encountered in ANC systems which must be taken into account. Virtual secondary path concept can be employed to deal with nonlinear secondary path. Using the filtered error based algorithm developed in [62] the adjoint virtual secondary path filter coefficient vector is defined as

$$\tilde{\mathbf{b}}_-(n) = [\tilde{b}_{M-1}(n), \tilde{b}_{M-2}(n-1), \dots, \tilde{b}_0(n-M+1)]$$

Output layer synaptic weights

Using adjoint virtual secondary path filter, the update equation for output layer synaptic weights is derived from (3.24) as follows

$$v_j(n+1) = v_j(n) + \mu u_j(n-M+1) \sum_{k=0}^{M-1} \tilde{b}_{M-1-k}(n-k) e(n-k) \quad j=0,1,\dots,J-1 \quad (3.31)$$

where $\tilde{b}_{M-1-k}(n-k)$ is the k^{th} coefficient of adjoint virtual secondary path filter at $(n-k)^{\text{th}}$ instant. The above equation can also be written as

$$v_j(n+1) = v_j(n) + \mu u_j(n-M+1)e'(n)$$

where $e'(n) = \tilde{\mathbf{b}}_-(n) * e(n)$

Hidden layer synaptic weights

The update equation for output layer synaptic weights is derived from (3.27) using adjoint virtual secondary path filter as follows

$$w_{ij}(n+1) = w_{ij}(n) + \mu v_j(n-M+1)F(c_j(n-M+1)) \sum_{k=0}^{M-1} \tilde{b}_{M-1-k}(n-k)e(n-k) \quad (3.32)$$

$$i=0,1,\dots,I-1 \text{ and } j=0,1,\dots,J-1$$

The above equation can also be written as

$$w_{ij}(n+1) = w_{ij}(n) + \mu v_j(n-M+1)F(c_j(n-M+1))e'(n) \quad (3.33)$$

$$i=0,1,\dots,I-1 \text{ and } j=0,1,\dots,J-1$$

3.5 Simulation and Results

Extensive simulation work has been done for various nonlinear situations, activation functions and some selected results are presented here to demonstrate the effectiveness and performance of the proposed algorithms. In all the experiments the mean-square error (MSE), defined by

$$MSE = 10 \log_{10} E(e^2(n)) \quad (3.34)$$

has been obtained through simulations in MATLAB 7.6.0 (R2008a) environment to assess the performance and validate the proposed algorithms. Here, $e^2(n)$ is the square of the error at n^{th} iteration, and $E(.)$ is the expectation operator. In each of the experiments, fifty independent trials are conducted and the average MSE(dB) is computed to obtain smoother convergence characteristics.

3.5.1 Experiment I

The first experiment considered a nonlinear ANC with nonlinear secondary path. The nonlinear primary path from noise source to error microphone is considered as in[62]

$$d(n) = 0.8x(n-6) + 0.6x(n-7) - 0.2x(n-8) - 0.5x(n-9) - 0.1x(n-10) + 0.4x(n-11) - 0.05x(n-12) + x(n-6)^3 \quad (3.35)$$

The nonlinear secondary acoustic path from secondary source to error microphone considered is

$$y(n) = u(n) + 0.35u(n-1) + 0.09u(n-2) - 0.05u(n)u(n-1) + 0.4u(n)u(n-2) \quad (3.36)$$

The reference noise is generated by filtering a uniformly distributed white noise through a lowpass filter of order 10 and cutoff frequency 350Hz. Simulations are done for proposed NFXLMS algorithm and are compared with Volterra filtered-x LMS (VFXLMS) algorithm. VFXLMS algorithm is the standard algorithm for nonlinear ANC. The structure of the MLP chosen is 10-3-1. The memory size of the second-order adaptive Volterra filter is 10. The step size for NFXLMS algorithm is chosen to be 0.03. However for VFXLMS algorithm the step sizes chosen are 0.003 and 0.0003 for linear and nonlinear coefficients respectively. Average mean square error (MSE) of fifty independent run of the algorithm is plotted with respect to number of iteration. Three different nonlinear activation functions are used for hidden layer neurons where as for all the experiments output layer neuron has linear activation function. Fig. 3.5 shows the performance when the activation function of the neurons in the hidden layer is

$$u_j(n) = \frac{1 - e^{-c_j(n)}}{1 + e^{-c_j(n)}} \quad (\text{activation function-I}) \quad (3.37)$$

Fig.3.6 shows the performance when the activation function of the neurons in the hidden layer is

$$u_j(n) = \frac{1}{1 + e^{-c_j(n)}} \quad (\text{activation function-II}) \quad (3.38)$$

Fig. 3.7 shows the performance when the activation function of the neurons in the hidden layer is

$$u_j(n) = \frac{e^{c_j(n)} - e^{-c_j(n)}}{e^{c_j(n)} + e^{-c_j(n)}} \quad (\text{activation function-III}) \quad (3.39)$$

In all the experiments hidden layer synaptic weights and output layer synaptic weights of neural controller are initialized as uniformly distributed random number in the range -0.5 to 0.5. All the weights of Volterra filter are also initialized to uniformly distributed random number in the range -0.5 to 0.5. The steady state MSE(dB) obtained by NFXLMS and VFXLMS algorithms is -26dB and -22dB respectively. The proposed

NFXLMS algorithm yield lower steady state MSE(dB) compared to standard VFXLMS algorithm for all the activation functions.

3.5.2 Experiment II

The nonlinear primary acoustic path from noise source to error microphone and nonlinear secondary acoustic path from secondary source to error microphone are same as that of simulation-I. The reference noise, MLP structure and weight initialization scheme also remain same as experiment-1 but the adaption algorithms are based on filtered-error method. MSE plot for NFELMS algorithm and VFELMS algorithm are shown in fig. 3.8, fig. 3.9 and fig. 3.10 for the three activation functions respectively. The step size for NFELMS algorithm is 0.03, while for VFELMS algorithm are 0.003 and 0.0003 for linear coefficients and nonlinear coefficients respectively. The steady state MSE(dB) obtained by NFELMS and VFELMS algorithms are -22dB and -18dB respectively. The proposed NFELMS algorithm yield much lower steady state MSE(dB) compared to VFELMS for all the activation functions.

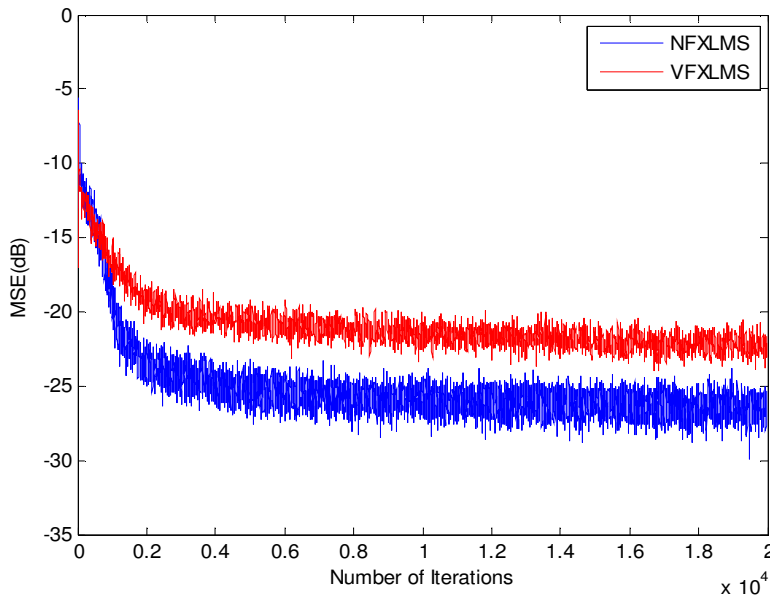


Fig. 3.5 MSE(dB) plot for NFXLMS and VFXLMS algorithms (activation function-I).

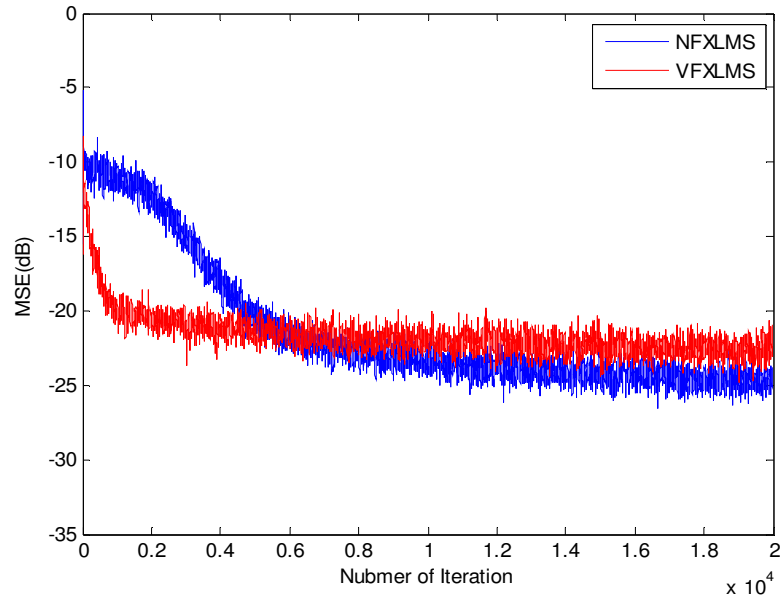


Fig. 3.6 MSE(dB) plot for NFXLMS and VFXLMS algorithms (activation function-II).

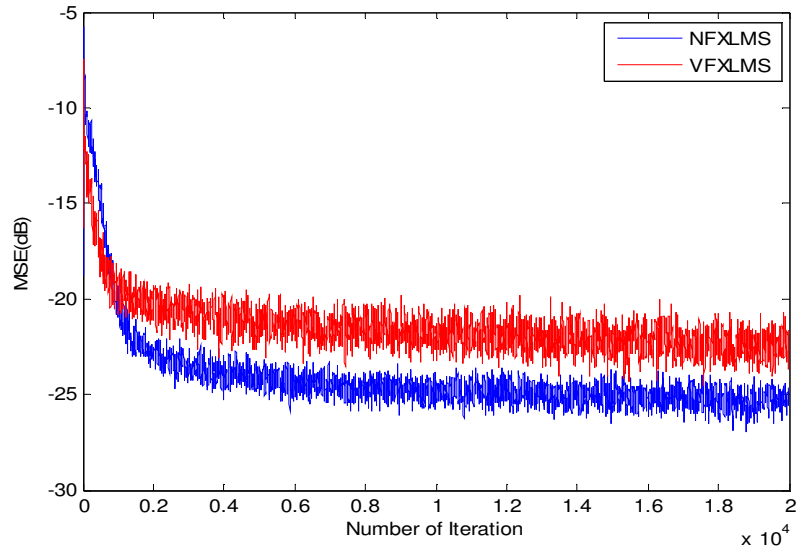


Fig. 3.7 MSE(dB) plot for NFXLMS and VFXLMS algorithms (activation function-III).

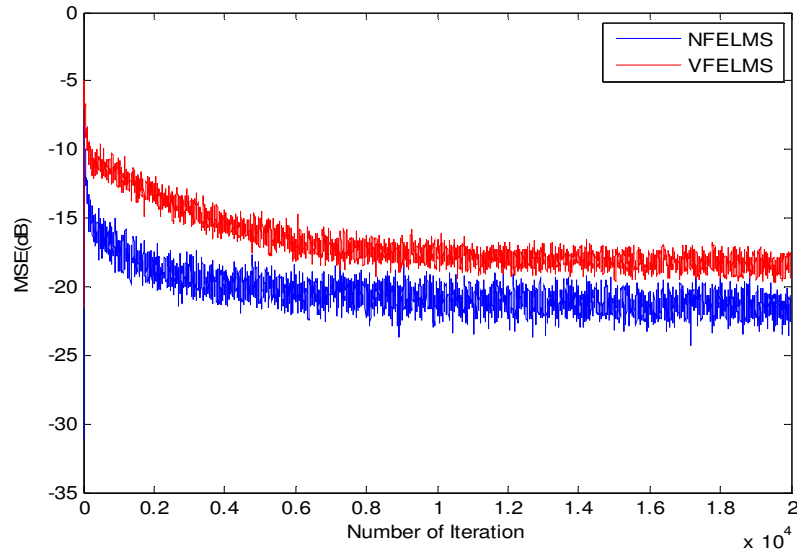


Fig. 3.8 MSE(dB) plot for NFELMS and VFELMS algorithms (activation function I).

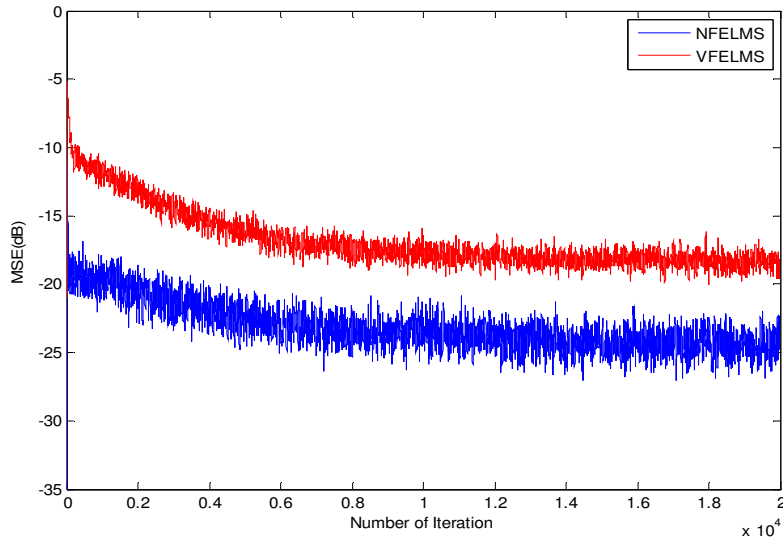


Fig. 3.9 MSE(dB) plot for NFELMS and VFELMS algorithms (activation function-II).

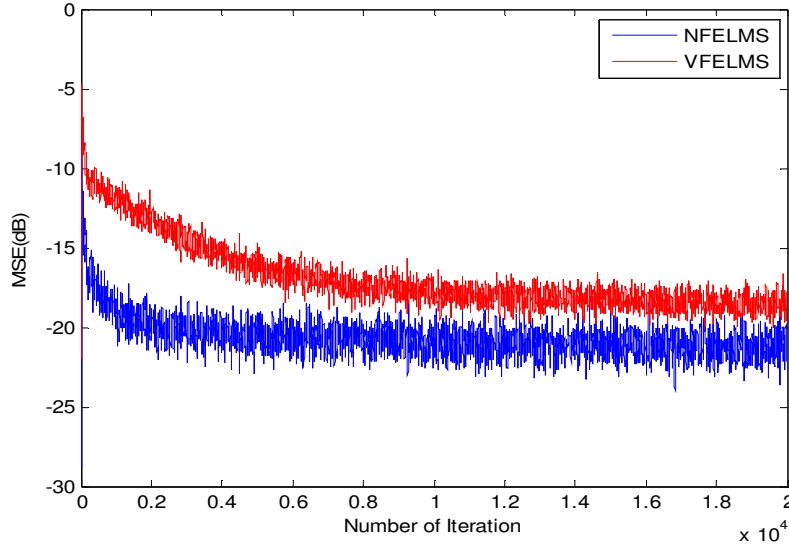


Fig. 3.10 MSE(dB) plot for NFELMS and VFELMS algorithms (activation function-III).

3.5.3 Experiment III

The reference noise signal is chosen to be a logistic chaotic type, which is generated using the recursive equation [64]

$$x(n+1) = \lambda x(n) [1 - x(n)] \quad (3.40)$$

where $\lambda = 4$ and $x(0) = 0.9$ are chosen. This nonlinear noise process is then normalized to have unity signal power. In this experiment the primary path transfer function considered is

$$A(z) = z^{-5} - 0.3z^{-6} + 0.2z^{-7} \quad (3.41)$$

and the secondary path transfer function is chosen to be the non minimum-phase model defined by

$$B(z) = \hat{B}(z) = z^{-2} + 1.5z^{-3} - z^{-4} \quad (3.42)$$

Simulations are done for proposed NFXLMS algorithm and are compared with VFXLMS algorithm. The structure of the MLP chosen is 10-3-1. The memory size of the second-order adaptive Volterra filter is 10. The step size for NFXLMS algorithm is 0.02 and for VFXLMS algorithm are 0.003 and 0.0003 for linear and nonlinear coefficients

respectively. Average mean square error (MSE(dB)) of fifty independent runs of the algorithms are plotted with respect to number of iteration. Similar to experiment-I, three different nonlinear activation functions are used for hidden layer neurons where as output neuron has linear activation function. Fig. 3.11, fig. 3.12 and fig. 3.13 show the performance of ANC for activation function I, II, III respectively. The steady state MSE(dB) obtained for NFXLMS and VFXLMS algorithms are -25dB and -22dB respectively.

3.5.4 Experiment IV

The primary path from noise source to error microphone and secondary path from secondary source to error microphone are same as that of experiment-III. The reference noise, MLP structure and weight initialization scheme also remain same as experiment-III but the adaption algorithms are based on filtered-error method. MSE plot for NFELMS algorithm and VFELMS algorithm are shown in fig. 3.14, fig. 3.15 and fig. 3.16 for the three activation function respectively. The step size for NFELMS algorithm is 0.03, and for VFELMS algorithm are 0.003 and 0.0003 for linear coefficients and nonlinear coefficients respectively. The steady state MSE(dB) obtained for NFELMS and VFELMS algorithms is -21dB.

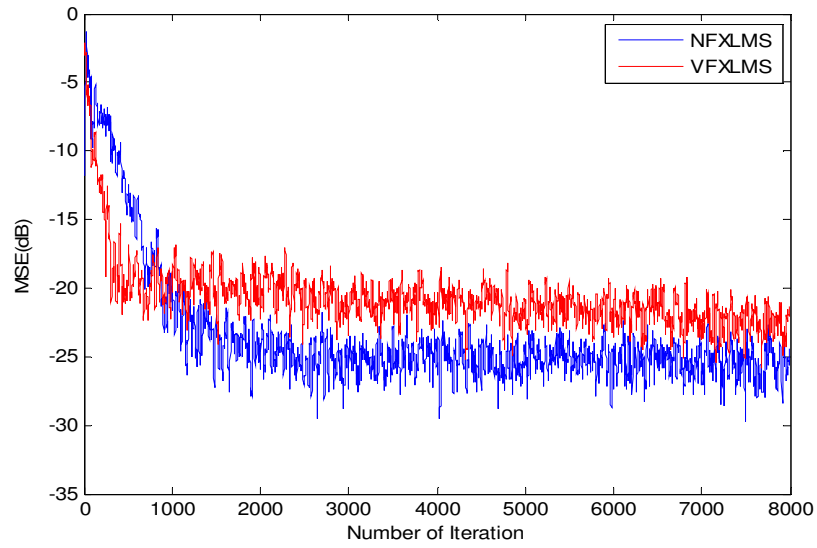


Fig. 3.11 MSE(dB) plot for NFXLMS and VFXLMS algorithms (activation function-I).

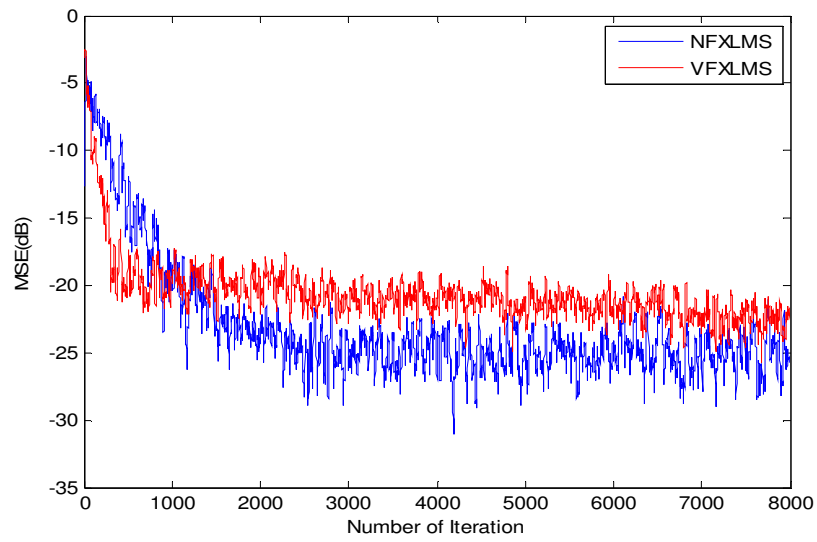


Fig. 3.12 MSE(dB) plot for NFXLMS and VFXLMS algorithms (activation function-II).

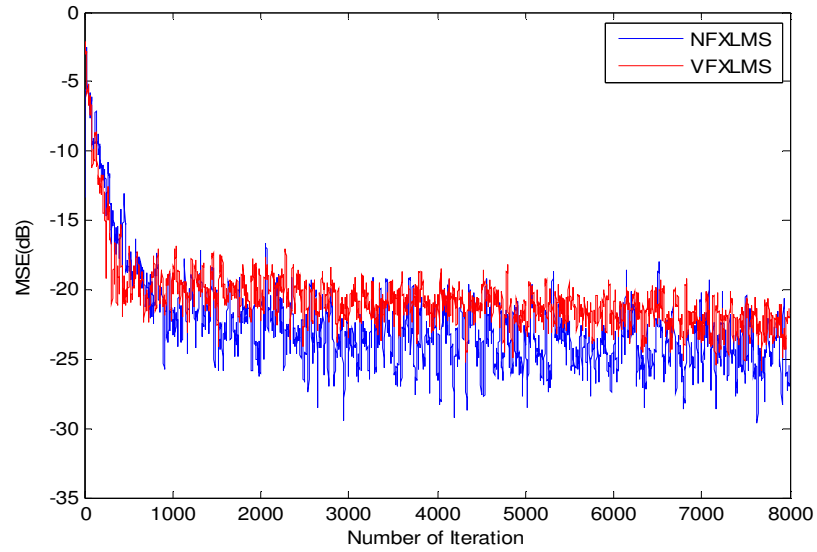


Fig. 3.13 MSE(dB) plot for NFXLMS and VFXLMS algorithms (activation function-III).

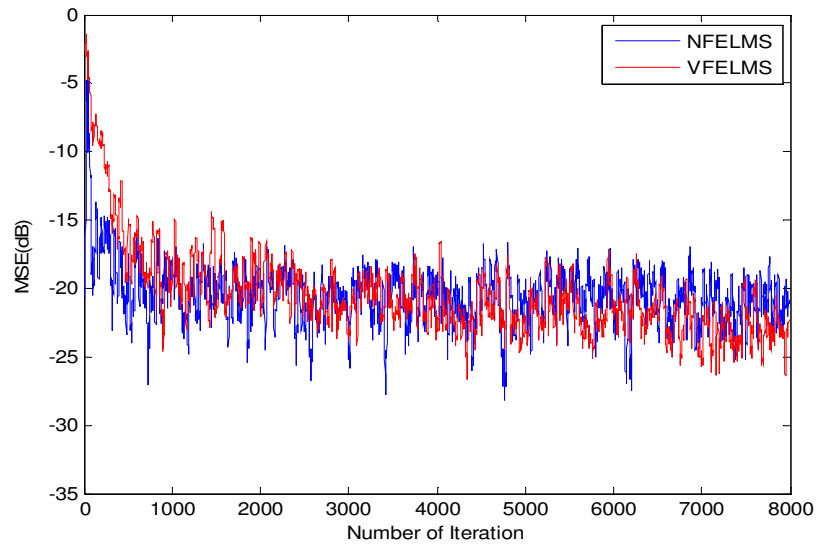


Fig. 3.14 MSE(dB) plot for NFELMS and VFELMS algorithms (activation function-I).

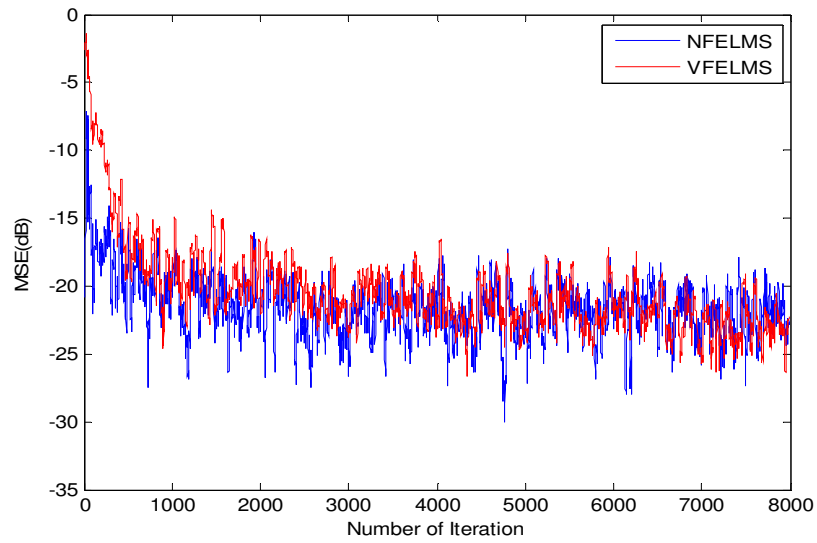


Fig. 3.15 MSE(dB) plot for NFELMS and VFELMS algorithms (activation function-II).

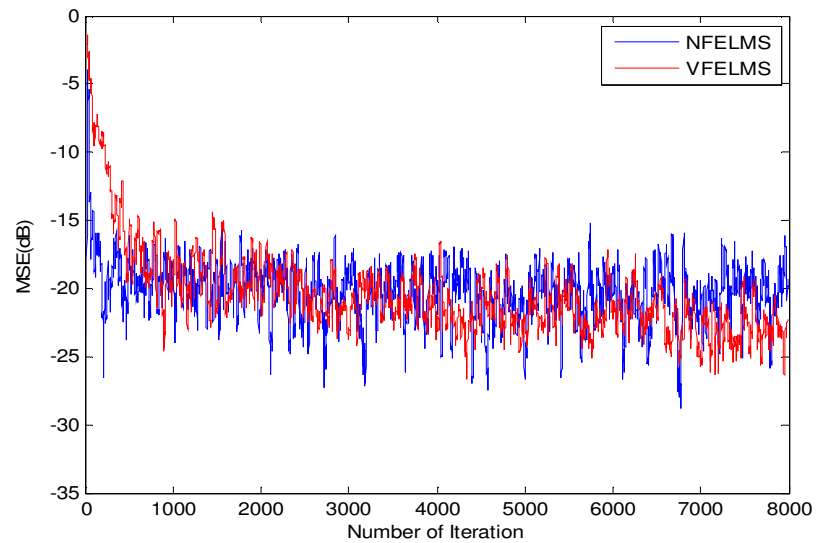


Fig. 3.16 MSE(dB) plot for NFELMS and VFELMS algorithms (activation function-III).

3.5.5 Experiment on Real Time Signals

To evaluate the performance of the proposed ANC structures and update algorithms in practical situations all the above experiments are repeated on some real time signals. Noise data from a Signal Processing Information Base (SPIB) are used. SPIB database have been provided by the Rice University [79]. The first signal is factory floor noise acquired by recording samples from 1/2" B&K condenser microphone on to digital audio tape (DAT). This noise was recorded near plate-cutting and electrical welding equipment. The second signal considered is a Buccaneer Jet cockpit noise. Buccaneer noise acquired by recording samples from 1/2" B&K condenser microphone onto digital audio tape (DAT). The Buccaneer was moving at a speed of 450 knots, and an altitude of 300 feet. The sound level during the recording process was 116 dBA. The detail information about both the noise are given below and are also plotted in fig. 3.17 and fig. 3.18 respectively.

Sampling rate :19.98 KHz

A/D: 16 bit

Pre-filter : Anti aliasing filter

Pre-emphasis : None

Filter : None

Duration-235sec

Length (uncompressed) : approx 9Mb

Taken from:

NOISE-ROM-0 signal.021

NATO: AC243/(Panel 3)/RSG-10

ESPRIT: Project No. 2589-SAM

Produced by:

Institute for Perception-TNO, The Netherlands

Speech Research Unit, RSRE, United Kingdom

Copyright:

TNO, Soesterberg, The Netherlands, Feb 1990

For more information:

Institute for Perception-TNO,

PO-box 23,
3769 ZG Soesterberg,
The Netherlands.

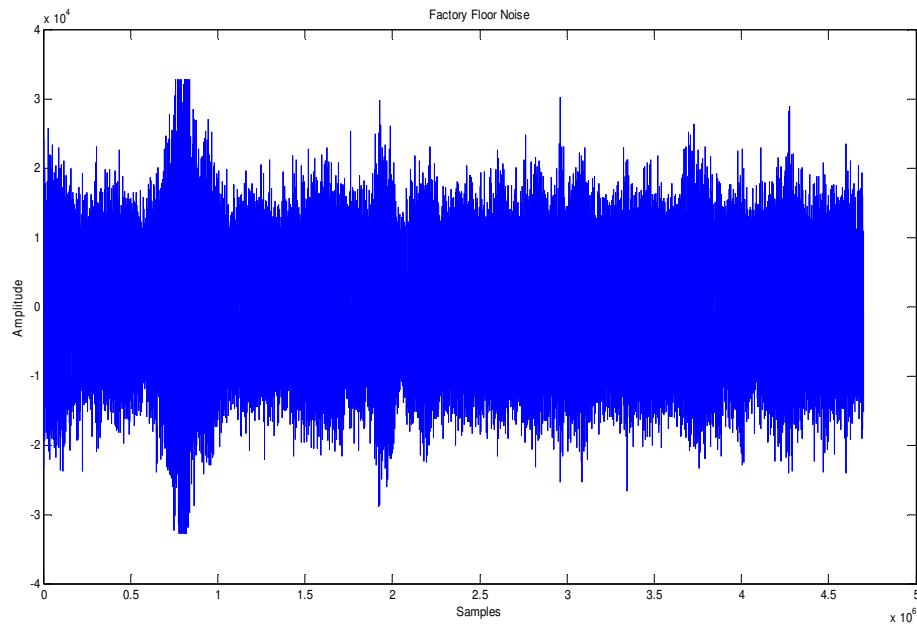


Fig. 3.17 Factory Floor Noise.

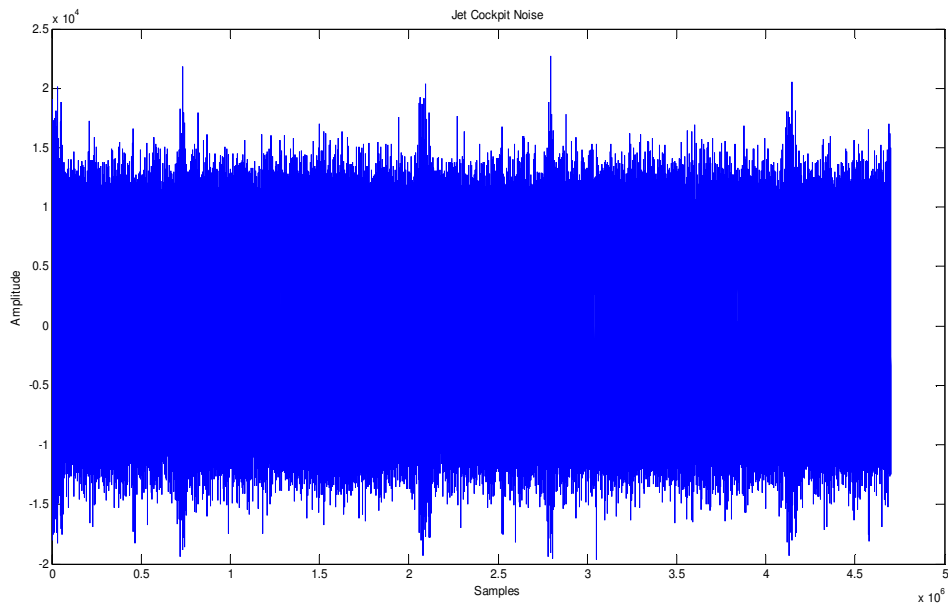


Fig. 3.18 Buccaneer Jet Cockpit Noise.

3.5.6 Experiment V (Real time signal)

In this experiment two real life noise signals are considered. The primary path, secondary path, neural network structure and adaptive Volterra filter structure same as that of experiment-I. Activation functions for hidden layer neurons are activation function-I where as output neuron has linear activation function. Fig. 3.19 and fig. 3.20 show the MSE(dB) plot of the proposed NFXLMS and VFXLMS algorithms for factory floor noise and Buccaneer jet cockpit noise respectively. The step size in case of factory floor noise for NFXLMS algorithm is 0.05 and for VFXLMS algorithm are 0.003 and 0.0003 for linear and nonlinear coefficients respectively. The step size in case of Buccaneer jet cockpit noise for NFXLMS algorithm is 0.01 and for VFXLMS algorithm and are 0.001 and 0.0001 for linear and nonlinear coefficients respectively. The steady state MSE(dB) obtained for NFXLMS and VFXLMS algorithms with factory floor noise are -28dB and -25dB and with Buccaneer jet cockpit noise are -25dB and -22 dB .

3.5.7 Experiment VI (Real time signal)

The proposed NFELMS and VFELMS algorithms are tested on two noise data collected from real world environment. MSE(dB) plots for factory floor noise and Buccaneer jet cockpit noise are shown in fig. 3.21 and fig. 3.22 respectively. The primary path and secondary path, neural network structure and adaptive Volterra filter structure are same as that of experiment-II. Activation functions for hidden layer neurons are activation function-I where as output neuron has linear activation function. The step size in case of factory floor noise for NFELMS algorithm is 0.04 and for VFELMS algorithm are 0.003 and 0.0003 for linear and nonlinear coefficients respectively. The step size in case of Buccaneer jet cockpit noise for NFELMS algorithm is 0.01 and for VFELMS algorithm are 0.001 and 0.0001 for linear and nonlinear coefficients respectively. The steady state MSE(dB) obtained for NFELMS and VFELMS algorithms with factory floor noise are -27dB and -20dB and with Buccaneer jet cockpit noise are -23dB and -18 dB respectively.

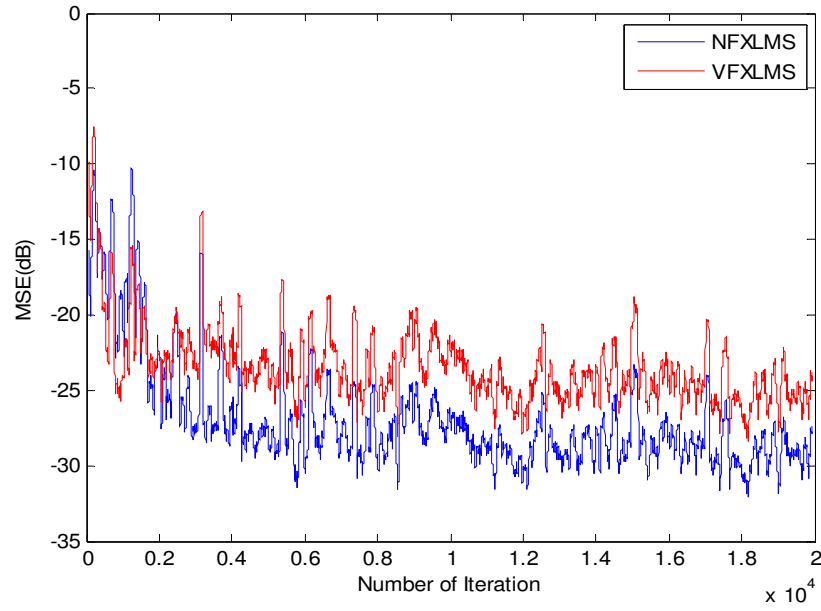


Fig. 3.19 MSE(dB) plot for NFXLMS and VFXLMS algorithms(Factory Floor Noise).

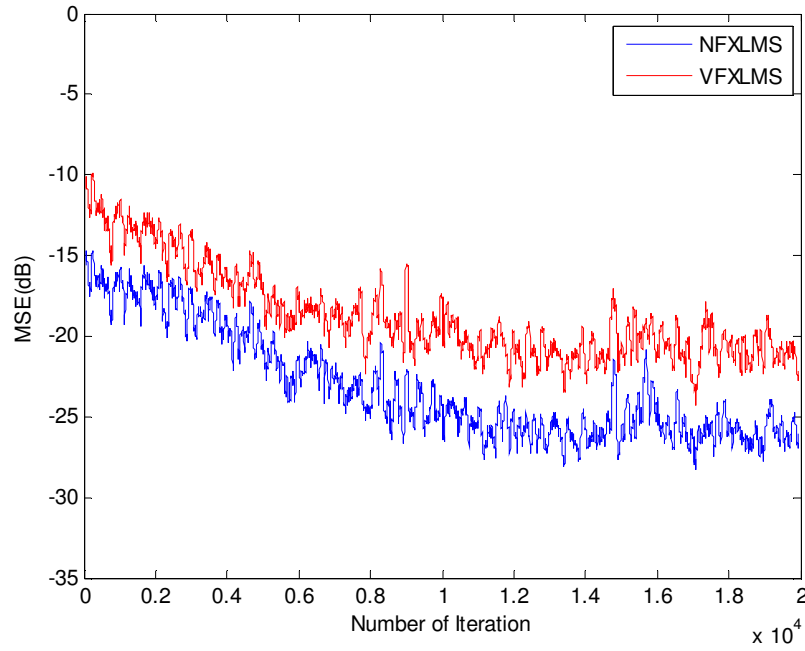


Fig. 3.20 MSE(dB) plot for NFXLMS and VFXLMS algorithms(Jet Cockpit Noise).

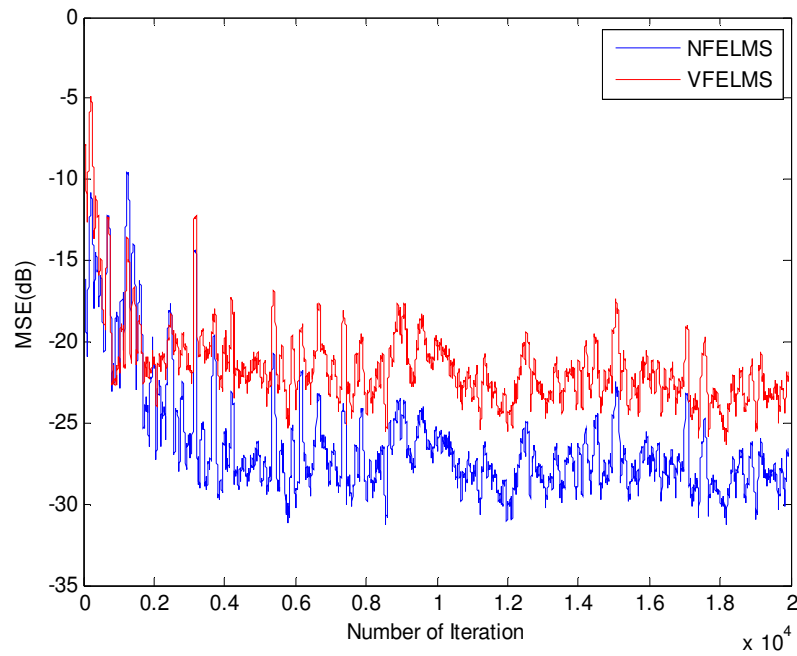


Fig. 3.21 MSE(dB) plot for NFELMS and VFELMS algorithms(Factory Floor Noise).

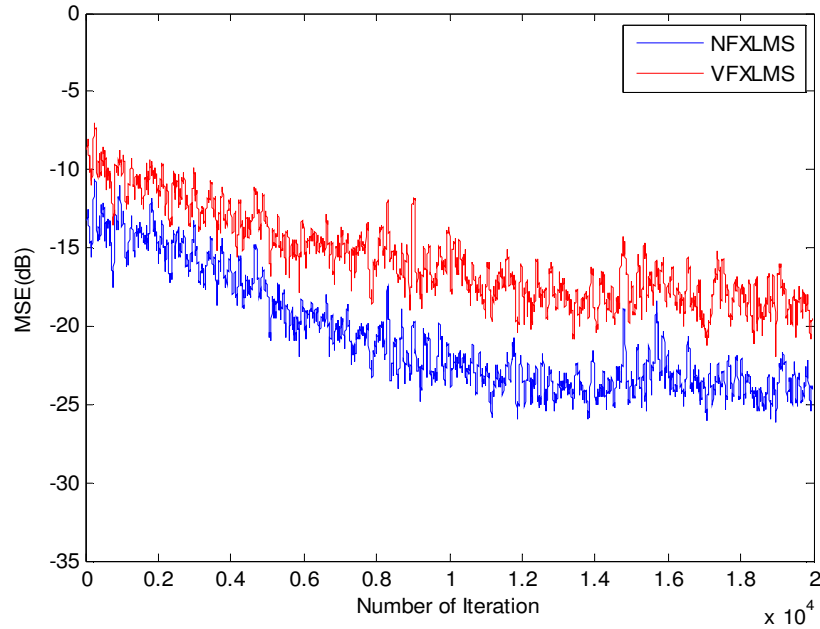


Fig. 3.22 MSE(dB) plot for NFELMS and VFELMS algorithms(Jet Cockpit Noise).

3.5.8 Experiment VII (Real time signal)

This experiment is conducted considering two real time signals as reference noise. The primary path and secondary path, neural network structure and adaptive Volterra filter structure are same as that of experiment-III. Activation functions for hidden layer neurons are activation function-I where as output neuron has linear activation function. The step size in case of factory floor noise for NFXLMS algorithm is 0.05 and for VFXLMS algorithm are 0.003 and 0.0003 for linear and nonlinear coefficients respectively. The step size in case of Buccaneer jet cockpit noise for NFXLMS algorithm is 0.01 and for VFXLMS algorithm are 0.001 and 0.0001 for linear and nonlinear coefficients respectively. Fig. 3.23 and fig. 3.24 show the MSE(dB) plot of the proposed NFXLMS and VFXLMS algorithms for factory floor noise and Buccaneer jet cockpit noise respectively. The steady state MSE(dB) obtained for NFXLMS and VFXLMS algorithms with factory floor noise are -30dB and -26dB and with Buccaneer jet cockpit noise are -33dB and -27 dB .

3.5.9 Experiment VIII (Real time signal)

In this experiment two real time reference signals are considered. The primary path and secondary path, neural network structure and adaptive Volterra filter structure are same as that of experiment-IV. MSE(dB) plots for factory floor noise and Buccaneer jet cockpit noise are shown in fig. 3.25 and fig. 3.26 respectively. The step size in case of factory floor noise for NFELMS algorithm is 0.04 and for VFELMS algorithm are 0.003 and 0.0003 for linear and nonlinear coefficients respectively. The step size in case of Buccaneer jet cockpit noise for NFELMS algorithm is 0.01 and for VFELMS algorithm are 0.001 and 0.0001 for linear and nonlinear coefficients respectively. The steady state MSE(dB) obtained for NFELMS and VFELMS algorithms with factory floor noise are -26dB and -22dB and with Buccaneer jet cockpit noise are -33dB and -27 dB respectively.

In all the experiments, considering real time signals as reference noise, it has been observed that the proposed neural based algorithms exhibit lower steady state MSE(dB) performance compared to standard Volterra based algorithms with comparable convergence time.

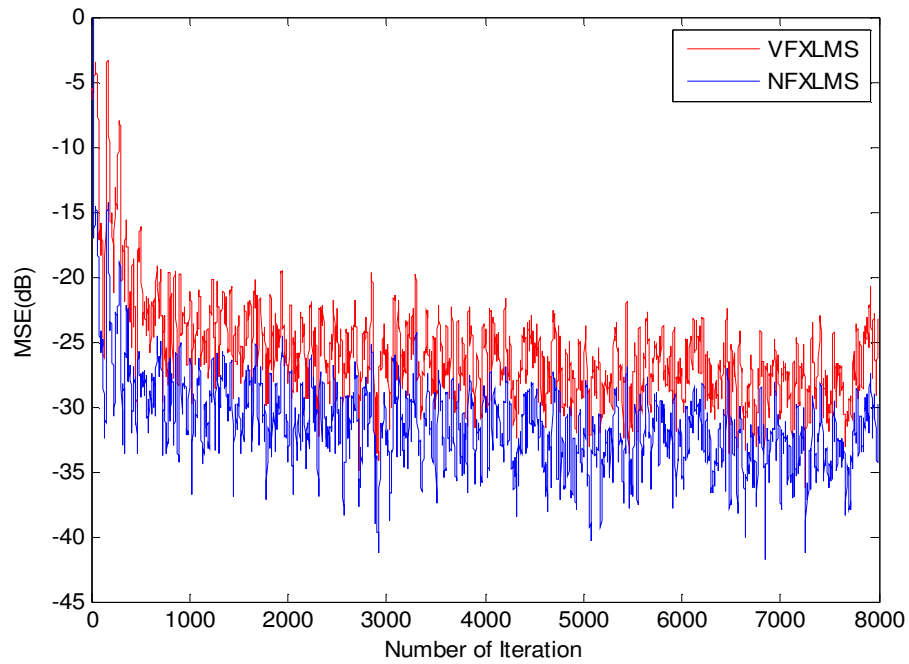


Fig. 3.23 MSE(dB) plot for NFXLMS and VFXLMS algorithms(Factory Floor Noise).

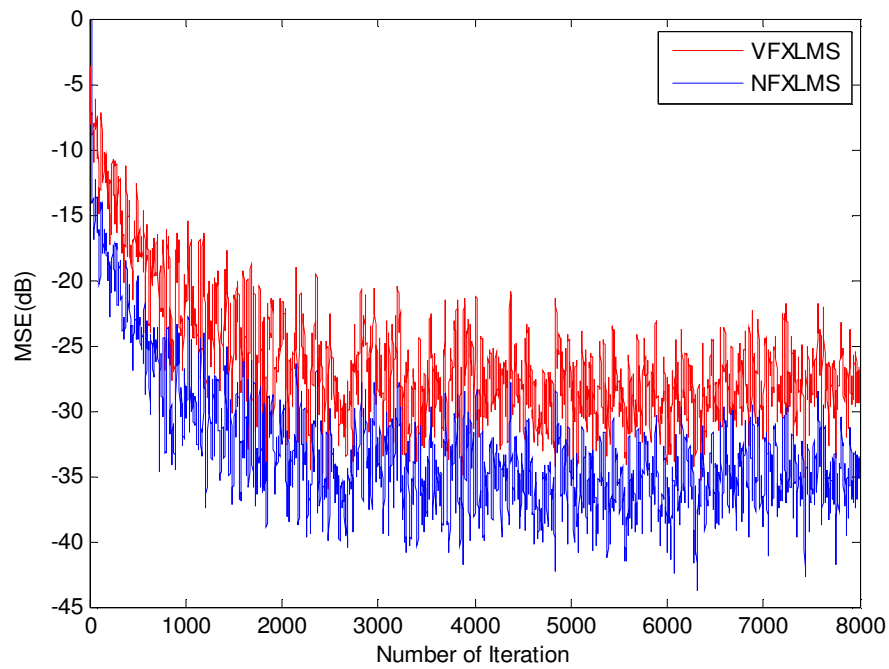


Fig. 3.24 MSE(dB) plot for NFXLMS and VFXLMS algorithms(Jet Cockpit Noise).

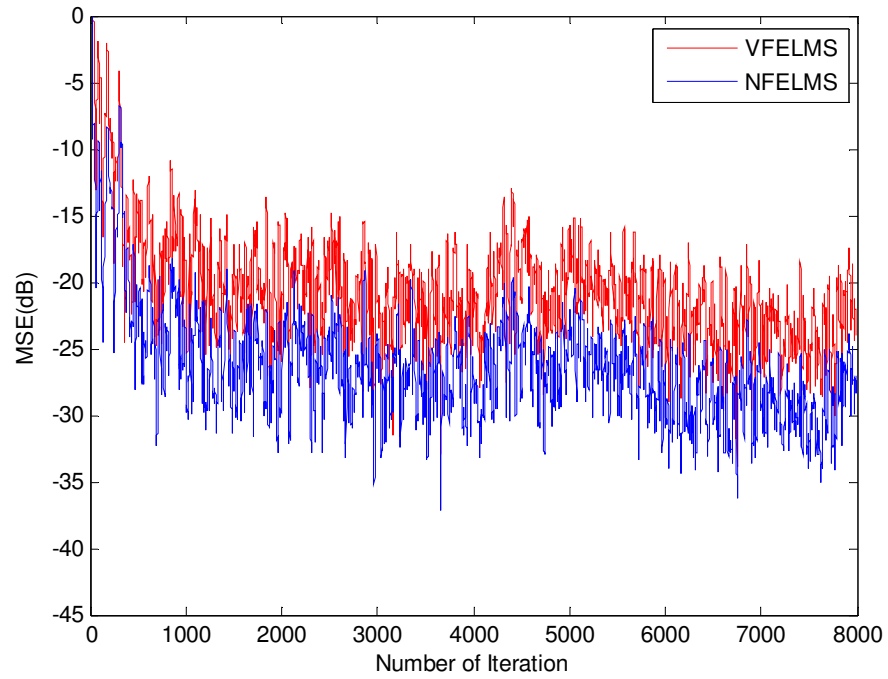


Fig. 3.25 MSE(dB) plot for NFELMS and VFELMS algorithms(Factory Floor Noise).

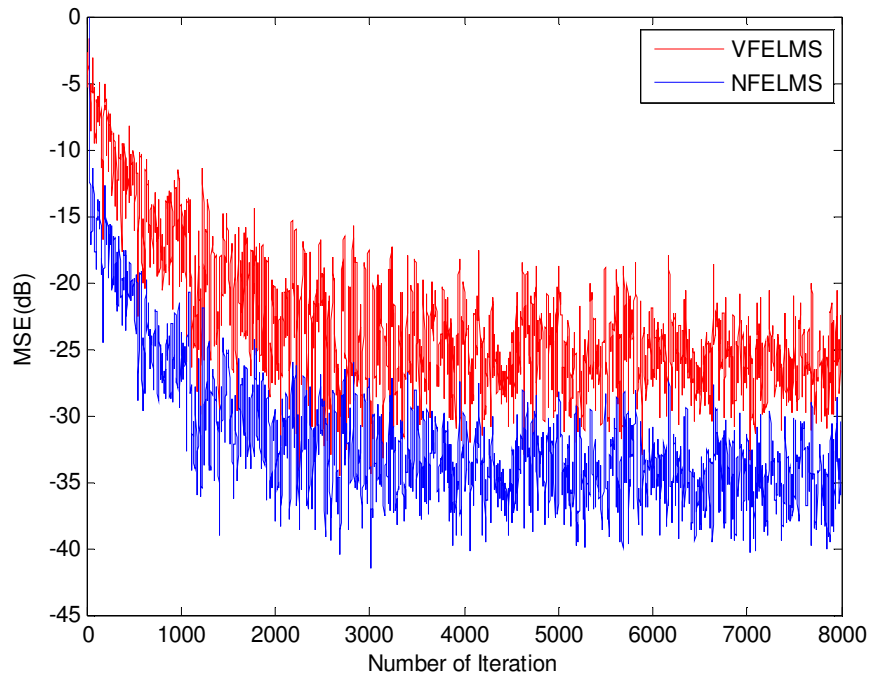


Fig. 3.26 MSE(dB) plot for NFELMS and VFELMS algorithms(Jet Cockpit Noise).

6 Summary

This chapter focuses on developing a MLP based neural network controller for nonlinear ANC. Separate algorithms were developed for nonlinear ANC, firstly when secondary path is assumed linear and then secondly when secondary path is assumed nonlinear. When secondary path is nonlinear, the developed algorithm is modified using virtual secondary path filter concept. By computer simulation it has been found that the developed algorithms are performing well for both linear secondary path and nonlinear secondary path. The performance of developed algorithms is also evaluated for real time reference signals. The proposed algorithms outperformed VFXLMS algorithm in terms of steady state MSE(dB). In order to take advantage of low computational complexity offered by filtered error LMS algorithm, both the developed algorithms are suitably modified. Performance of the modified algorithms are also analyzed from computer simulations and compared with that of VFELMS algorithm. The neural based ANC have resulted in improved performance in comparison to the Volterra based ANCs.

CHAPTER 4

Legendre Neural Network for Nonlinear Active Noise Control

4.1 Background

Based on the rapid progress of high-speed and low-cost computing devices such as digital signal processor (DSP), digital active noise control (ANC) techniques have been receiving much attention because of the better performance over conventional passive methods. Passive methods of noise control are effective over a broad frequency range except at the lower end (below 600 Hz). On a contrary active noise control is suitable for low frequency noise only and it is not implementable for higher frequencies. So very often passive and active noise control methods are employed simultaneously in order to get an overall noise suppression. In some applications where only low frequency noise is present employing only ANC is sufficient.

Presence of inherent nonlinearity in the ANC makes the antinoise generation process very complex. In [35] L. Tan and J. Jiang proposed a nonlinear ANC using adaptive Volterra filter and developed Volterra FXLMS (VFXLMS) algorithm. They demonstrate that the developed algorithm can improve control performance over the linear standard filtered-x LMS algorithm under the following conditions.

- i) The reference noise sensed by a reference microphone is a nonlinear and predictable noise process, while the secondary path transfer function of an ANC system has nonminimum phase.
- ii) The primary path exhibits nonlinear behavior

The drawback of this algorithm is high computational complexity. In [34] L. Tan and J. Jiang proposed a truncated second order Volterra structure for NANC which is computationally more efficient.

Y. H. Pao [63] proposed an alternate neural network structure called functional link artificial neural network (FLANN) with an object to reduce the training time of the neural network and to improve convergence speed. Unlike MLANN where the nodes have nonlinear activation functions, in case of FLANN the links have nonlinear functions. In [64] D. P. Das and G. Panda employed FLANN to develop filtered-s LMS (FSLMS) algorithm (here this algorithm is termed as FLANN-FXLMS or FFXLMS algorithm) for nonlinear ANC. FFXLMS algorithm outperform VFXLMS algorithm in terms of steady state mean square error and has less computational requirement. D. P. Das and G. Panda [64] also proposed fast FFXLMS algorithm with a objective to reduce computational requirement. In [62] D. Zhou and V. DeBrunner compared the VFXLMS algorithm and FFXLMS algorithm and represented both by a generalized function expansion equation. They have extended the algorithms to deal with nonlinear ANC with nonlinear secondary path (NSP) by introducing virtual secondary path concept. They have also introduced adjoint virtual secondary path to employ filtered-e based algorithms. Block oriented (such as, Wiener, Hammerstein and Linear-Nonlinear-Linear structure) representation of nonlinear secondary path was discussed by them. Basically algorithms for nonlinear ANC can now be classified as nonlinear ANC with linear secondary path (LSP) and nonlinear ANC with nonlinear secondary path (NSP).

In this chapter Legendre neural network (LNN) is used for nonlinear ANC. The adaptive algorithm for Legendre neural network is also developed. The algorithm is found to be simple and easy to implement and have low computational complexity. In order to reduce computational complexity further a reduced structure Legendre neural network has been proposed. The reduced structure Legendre neural network reduces the computational complexity without sacrificing the performance. Weight update algorithms for reduced structure LNN based on Filtered-x least mean square (FXLMS), Filtered-e least mean square (FELMS) and Filtered-x recursive least square (FXRLS) are developed. These are named as reduced structure LFXLMS algorithm, LFELMS algorithm and LFXRLS algorithm respectively. These algorithms require less computation compared to FFXLMS algorithm and are modified to deal with NSP which rests upon virtual secondary path concept. [66]-[72] reported different strategies to derive fast LMS algorithms. Based on these papers, in order to reduce computational complexity further, a faster version of

reduced structure LFXLMS algorithm is also developed this chapter in which unlike LFXLMS algorithm, weights are updated in every alternative iterations.

4.2 Reduced Structure Legendre Neural Network for Nonlinear ANC

4.2.1 Legendre Polynomial

The Legendre polynomials are denoted by $L_p(x)$, where $p=0,1,2,\dots,P$, P is the order of expansion and x is the argument of the polynomial. $L_p(x)$ constitute a set of orthogonal polynomials as solutions to the differential equation

$$\frac{d}{dx} \left[(1-x^2) \frac{dy}{dx} \right] + n(n+1)y = 0.$$

The zero and the first order Legendre polynomials are, respectively, given by

$$L_0(x) = 1$$

and $L_1(x) = x$.

The higher order polynomials are given by

$$L_2(x) = \frac{1}{2}(3x^2 - 1)$$

$$L_3(x) = \frac{1}{2}(5x^3 - 3x)$$

$$L_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3)$$

...

The recursive formula to generate higher order Legendre polynomials is expressed as

$$L_{p+1}(x) = \frac{1}{(n+1)} [(2n+1)xL_p(x) - nL_{p-1}(x)] \quad (4.1)$$

Some of the important properties of Legendre polynomials are that (i) they are orthogonal polynomials, (ii) they arise in numerous problems especially in those involving spheres or spherical coordinates or exhibiting spherical symmetry and (iii) in spherical polar coordinates, the angular dependence is always best handled by spherical harmonics that are defined in terms of Legendre functions.

In case of an ANC Legendre polynomial of a reference noise signal sample $x(n)$ is computed and rearranged to form a vector as follows

$$L(x(n)) = [1, x(n), \frac{1}{2}(3x^2(n) - 1), \frac{1}{2}(5x^3(n) - 3x(n)), \dots, \frac{1}{(n+1)}\{(2n+1)x(n)L_p(x(n)) - nL_{p-1}(x(n))\}]$$

Reference noise signal vector at n^{th} instant consist of $x(n)$ and it's delayed samples and is defined as $\mathbf{x}(n) = [x(n), x(n-1), x(n-2), \dots, x(n-N+1)]$

Legendre polynomial of $\mathbf{x}(n)$ can be obtained by expanding each sample of $\mathbf{x}(n)$ as

$$\begin{aligned} L(\mathbf{x}(n)) = & [1, x(n), \frac{1}{2}3x^2(n) - 1, \dots \\ & 1, x(n-1), \frac{1}{2}3x^2(n-1) - 1, \dots \\ & 1, x(n-2), \frac{1}{2}3x^2(n-2) - 1, \dots \\ & \dots \\ & 1, x(n-N+1), \frac{1}{2}3x^2(n-N+1) - 1] \end{aligned} \quad (4.2)$$

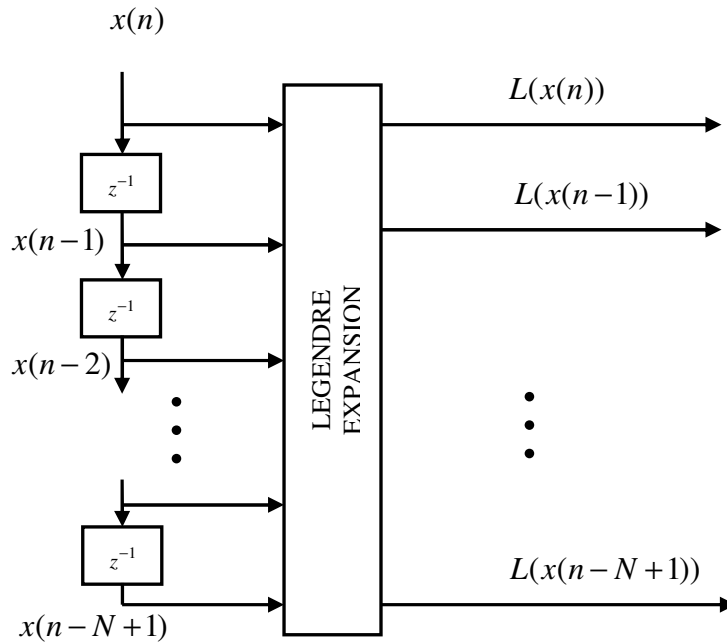


Fig. 4.1 Legendre polynomial expansion.

4.2.2 Legendre Neural Network

The elements of above Legendre polynomial expansion of $\mathbf{x}(n)$ are rearranged and partitioned to form P new vectors. Taking first element of each row from above equation a vectors $\mathbf{s}_0(n) = [1, 1, 1, 1, \dots]$ is formed. Taking the second element of each row from above equation a second vector $\mathbf{s}_1(n)$ is formed as shown below

$$\mathbf{s}_1(n) = [x(n), x(n-1), \dots, x(n-N+1)].$$

Similarly third vector is formed

$$\mathbf{s}_2(n) = [\frac{1}{2}(3x^2(n)-1), \frac{1}{2}(3x^2(n-1)-1), \dots, \frac{1}{2}(3x^2(n-N+1)-1)].$$

Similar manner P new vector, $\mathbf{s}_0(n), \mathbf{s}_1(n), \mathbf{s}_2(n), \dots, \mathbf{s}_p(n)$ can be formed. The vectors $\mathbf{s}_0(n), \mathbf{s}_1(n), \mathbf{s}_2(n), \dots, \mathbf{s}_p(n)$ represent $0^{\text{th}}, 1^{\text{st}}, 2^{\text{nd}}, \dots, P^{\text{th}}$ order Legendre expansion. Using these vectors Legendre Neural Network is formed as shown in fig.4.2 where $\mathbf{w}_0(n), \mathbf{w}_1(n), \dots, \mathbf{w}_p(n)$ are adaptable weight vectors. All the vectors can be combined to form a single vectors $\mathbf{s}(n)$.

$$\mathbf{s}(n) = [\mathbf{s}_0(n), \mathbf{s}_1(n), \mathbf{s}_2(n), \dots, \mathbf{s}_p(n)]$$

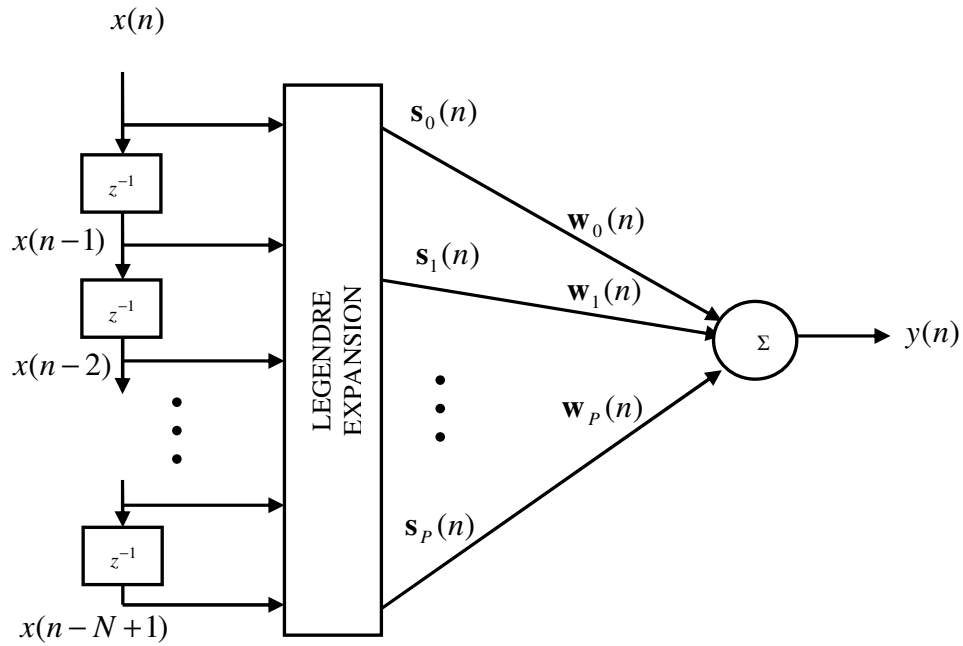


Fig. 4.2 Legendre neural network.

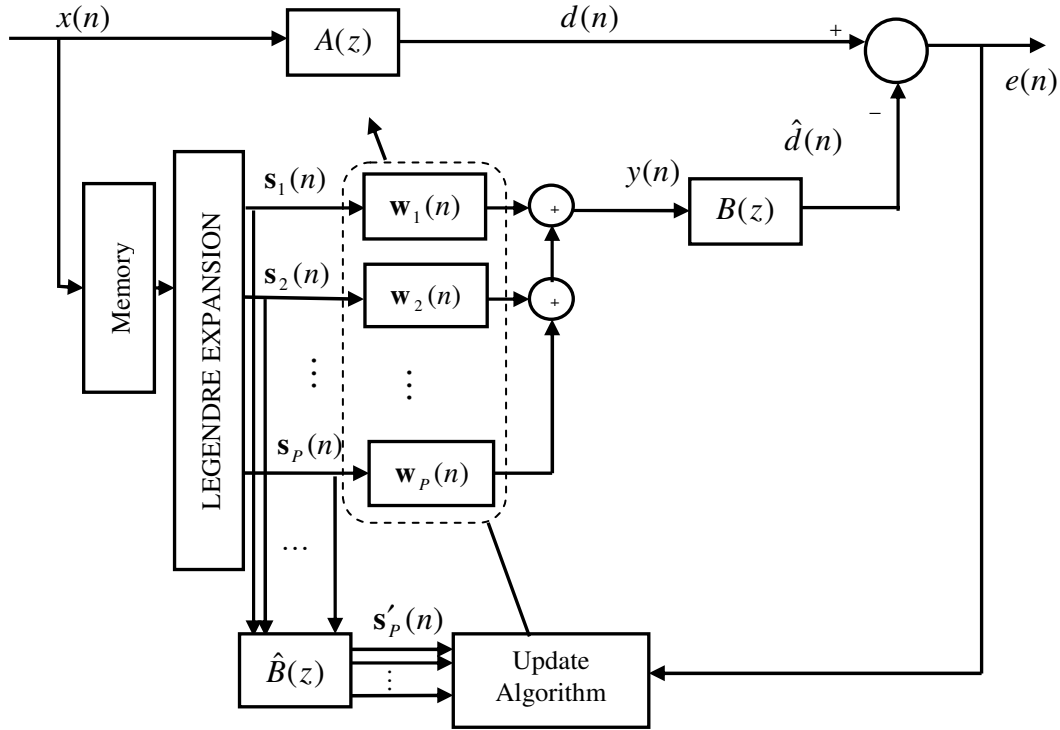


Fig. 4.3 Block diagram of ANC using reduced structure Legendre neural network.

4.3 LFXLMS Algorithm

Structure of the ANC using Legendre Neural Network is shown in fig. 4.3 where the input vector $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]$ is transformed into an output vector $\mathbf{s}(n)$ given by $\mathbf{s}(n) = L(\mathbf{x}(n))$. The nonlinear function $L(\mathbf{x}(n))$ represents a set of the orthogonal basis functions, implemented in the ‘‘Legendre expansion’’ block. Here the N -dimensional input pattern $\mathbf{x}(n)$ is enhanced to an $N(P+1)$ -dimensional enhanced pattern

$$\mathbf{s}(n) = [\mathbf{s}_0(n) \ \mathbf{s}_1(n) \ \dots \ \mathbf{s}_p(n)].$$

where

$$\mathbf{s}_0(n) = L_0(\mathbf{x}(n)) = [1 \ 1 \ \dots \ N \text{ number of } 1]$$

$$\mathbf{s}_1(n) = L_1(\mathbf{x}(n)) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]$$

$$\mathbf{s}_2(n) = L_2(\mathbf{x}(n)) = [\frac{1}{2}(3x^2(n)-1) \ \frac{1}{2}(3x^2(n-1)-1) \ \dots \ \frac{1}{2}(3x^2(n-N+1)-1)]$$

.....

Comparing to FLANN where trigonometric functions are used in the functional expansion, LNN uses Legendre orthogonal functions. The major advantage of LNN over FLANN is that the evaluation of Legendre polynomials involves less computation compared to that of the trigonometric functions. Therefore, LNN offers faster training compared to FLANN. Corresponding to $P+1$ expanded input vectors the network has $P+1$ number of adaptive filters $\mathbf{w}_0(n), \mathbf{w}_1(n), \dots, \mathbf{w}_P(n)$, operating in parallel. This approach is called filter bank implementation [36]. The order of each adaptive filter is N . Employing filter bank implementation output of LNN at time n is obtained by summing outputs of all the adaptive filters.

$$y(n) = \sum_{i=0}^P y_i(n) = \sum_{i=0}^P \mathbf{s}_i(n) \mathbf{w}_i^T(n) \quad (4.3)$$

where $\mathbf{w}_i(n)$ is the weight vector of i^{th} adaptive filter at n^{th} instant. Estimated desired signal $\hat{d}(n)$ is obtained by filtering LNN output by the estimated secondary path $B(z)$. Error at time n is defined as $e(n) = d(n) - \hat{d}(n)$. A popularly used cost function based on the mean-squared-error criterion is chosen here.

$$\begin{aligned} \xi(n) &= \frac{1}{2} [e^2(n)] \\ &= \frac{1}{2} [d(n) - \hat{d}(n)]^2 \end{aligned}$$

Using the FXLMS algorithm the weight vectors of each adaptive filter is updated as

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \mu e(n) \mathbf{s}'_i(n) \quad (4.4)$$

where $\mathbf{s}'_i(n)$ is the input signal, $\mathbf{s}_i(n)$, filtered through the estimated secondary path and μ is the step size which control convergence and stability. This algorithm is called Legendre FXLMS (LFXLMS) algorithm. In Legendre neural network for P^{th} order Legendre expansion $P+1$ number of adaptive filters operate in parallel. But it is observed that the input vector for first adaptive filter always contain 1, so it not dependant on

reference noise signal and it don't carry any information about the reference noise signal. Intuitively it is concluded that removing first adaptive filter from the network may not affect the network performance. This has been confirmed from extensive simulation work that removing the first adaptive filter did not degrade the network performance. This new structure is called as reduced structure Legendre neural network. The advantage of reduced structure Legendre neural network is to reduce the computational complexity. It has been seen that for 2nd order Legendre expansion, the saving in computational complexity is 33%, while for 3rd order expansion saving is 25%. From simulation study it has been observed that 3rd order expansion is sufficient to obtain noise reduction. Increasing the order of expansion does not result in further mean square error reduction. The output of reduced order Legendre neural network can now be written as

$$y(n) = \sum_{i=1}^P y_i(n) = \sum_{i=1}^P \mathbf{s}_i(n) \mathbf{w}_i^T(n)$$

In LFXLMS algorithm all the expanded input vectors have to be separately filtered through the estimated secondary path. Here also dropping the first input vector results reduction in number of filtering required so computational requirement is further reduced.

4.3.1 Nonlinear Secondary Path

For nonlinear active noise cancellation with nonlinear secondary path the update equation for weight vectors is written as

$$\begin{aligned} \mathbf{w}_i(n+1) &= \mathbf{w}_i(n) - \frac{1}{2} \mu \frac{\partial \xi(n)}{\partial \mathbf{w}_i(n)} \\ &= \mathbf{w}_i(n) + \mu E \left\{ e(n) \frac{\partial \hat{d}(n)}{\partial \mathbf{w}_i(n)} \right\} \quad \text{where } i=1, 2, \dots, P \end{aligned}$$

As in the classic LMS algorithm, we can use the instantaneous value to approximate the ensemble mean, yielding

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \mu e(n) \frac{\partial \hat{d}(n)}{\partial \mathbf{w}_i(n)}$$

Note that

$$\frac{\partial \hat{d}(n)}{\partial \mathbf{w}_i(n)} = \sum_{m=0}^{M-1} \frac{\partial \hat{d}(n)}{\partial y(n-m)} \cdot \frac{\partial y(n-m)}{\partial \mathbf{w}_i(n)}$$

where M is the memory size of the NSP. Assuming that for small step size, $\mathbf{w}_i(n)$ is slowly varying it can be written that

$$\frac{\partial y(n-m)}{\partial \mathbf{w}_i(n)} \approx \frac{\partial y(n-m)}{\partial \mathbf{w}_i(n-m)} \quad (4.5)$$

But $y(n-m) = \mathbf{w}_i^T(n-m)\mathbf{s}_i(n-m)$

Putting this value in (4.5) we get

$$\frac{\partial y(n-m)}{\partial \mathbf{w}_i(n)} = \mathbf{s}_i(n-m)$$

The update equation of $\mathbf{w}_i(n)$ can now be written as

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \mu e(n) \sum_{m=0}^{M-1} \frac{\partial \hat{d}(n)}{\partial y(n-m)} \mathbf{s}_i(n-m) \quad (4.6)$$

The first term of summation of (4.6), is found to be

$$\frac{\partial \hat{d}(n)}{\partial y(n-m)} = \left[\frac{\partial \hat{d}(n)}{\partial y(n)}, \frac{\partial \hat{d}(n)}{\partial y(n-1)}, \frac{\partial \hat{d}(n)}{\partial y(n-2)}, \dots, \frac{\partial \hat{d}(n)}{\partial y(n-M+1)} \right] \quad (4.7)$$

The elements of above vector are found to be time varying. This vector is called virtual secondary path, denoted by $\tilde{\mathbf{b}}(n)$ and defined as follows [62]

$$\tilde{\mathbf{b}}(n) = [\tilde{b}_0(n), \tilde{b}_1(n), \dots, \tilde{b}_{M-1}(n)]$$

Putting (4.7) in (4.6) the weight update equation now becomes

$$\begin{aligned} \mathbf{w}_i(n+1) &= \mathbf{w}_i(n) + \mu e(n) \sum_{m=0}^{M-1} \tilde{b}_m(n) \mathbf{s}_i(n-m) \\ \mathbf{w}_i(n+1) &= \mathbf{w}_i(n) + \mu e(n) \tilde{\mathbf{s}}_i'(n) \end{aligned} \quad (4.8)$$

where $\tilde{\mathbf{s}}_i'$ is the expanded input signal filtered through virtual secondary path. This update algorithm is LFXLMS algorithm for nonlinear secondary path.

4.4 LFELMS Algorithm

Adjoint LMS algorithm was developed by Wan [14] and provides a simple alternative to the FXLMS algorithms. In adjoint LMS, the error signal (rather than the input signal) is filtered through an adjoint secondary path filter. This algorithm is alternatively termed as filtered error least mean square (FELMS) algorithm. FELMS

algorithm drastically reduces computational complexity of FXLMS algorithm for multichannel ANC. The degree of saving in computational complexity increases with increase in number of channels. Saving in computational complexity can also be achieved for our LFXLMS algorithm by using the technique of FELMS algorithms. Structure of Legendre neural network is such that a number of adaptive filters operate in parallel which is called as filter bank approach. The key for the application of FELMS algorithm is to develop the adjoint secondary path. In case secondary path is linear one (LSP) adjoint secondary path is obtained by writing the coefficients of secondary path in reverse order. Thus adjoint secondary path can be written as follows

$$\mathbf{b}_{-}(n) = [b_{M-1}(n), b_{M-2}(n), \dots, b_0(n)] \quad (4.9)$$

The block diagram of secondary path filter and adjoint secondary path filter are shown in fig. 4.4 and fig. 4.5 respectively.

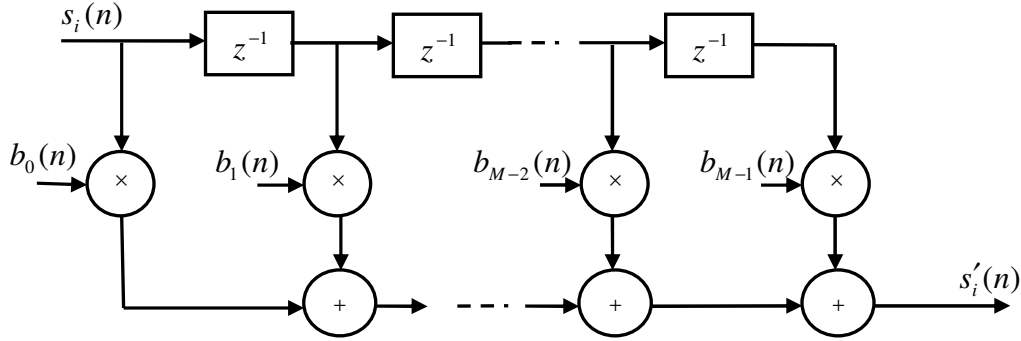


Fig. 4.4 Secondary path filter

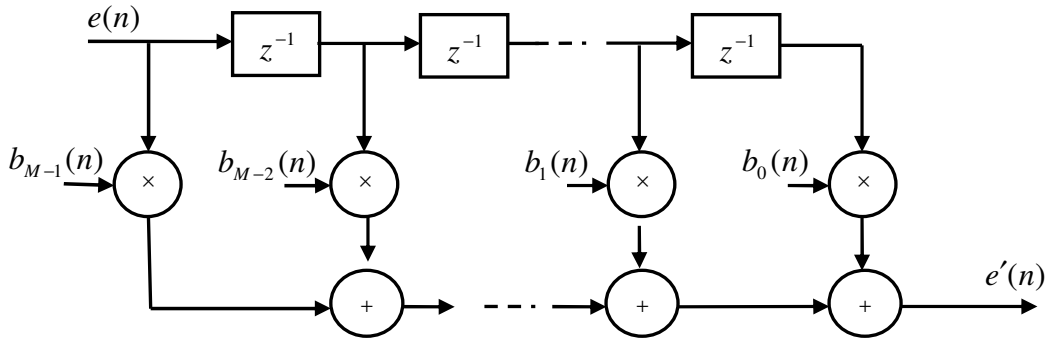


Fig. 4.5 Adjoint secondary path filter

When secondary path is nonlinear (NSP) the adjoint secondary path cannot be formed directly by writing the coefficients in reverse order. First the virtual secondary path filter is obtained and then adjoint virtual secondary path filter can be derived from it. Since virtual secondary path is a time varying filter so its adjoint version can be obtained by not only reversing the filter coefficients but also using delayed filter coefficients. The adjoint virtual secondary path is defined as follows [62]

$$\tilde{\mathbf{b}}_{-}(n) = [\tilde{b}_{M-1}(n) \quad \tilde{b}_{M-2}(n-1) \quad \dots \quad \tilde{b}_0(n-M+1)] \quad (4.10)$$

The block diagram of secondary path filter and adjoint secondary path filter are shown in fig. 4.6 and fig. 4.7 respectively.

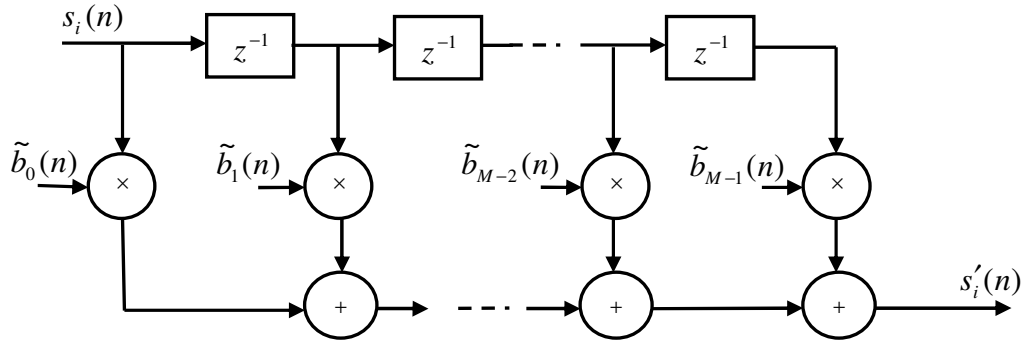


Fig. 4.6 Virtual secondary path filter

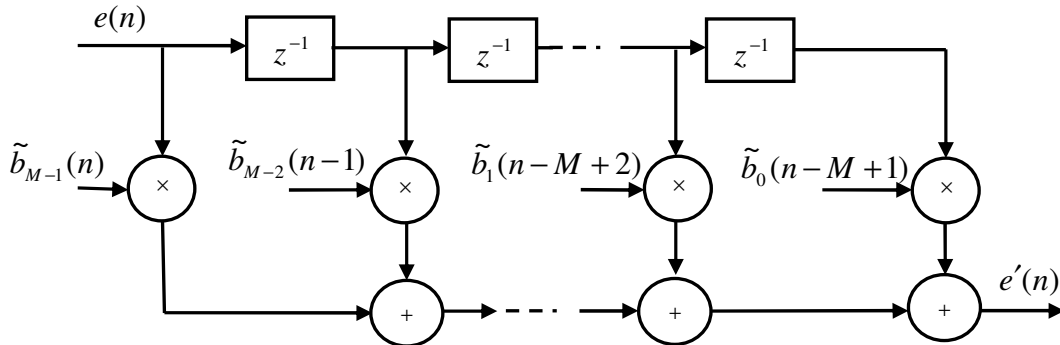


Fig. 4.7 Adjoint virtual secondary path filter

The weight update equation can be written as

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \mu e'(n) \mathbf{s}_i(n-M+1) \quad (4.11)$$

where $e'(n)$ is the error filtered through the adjoint secondary path for LSP and adjoint virtual secondary path for NSP. M is the length of virtual secondary path.

4.5 LFXRLS Algorithm

Generally recursive least square (RLS) algorithm is employed to enhance speed of convergence but at the cost of increase in computational complexity [7], [8]. RLS algorithm for Legendre neural network is used to develop Legendre filtered-x recursive least square (LFXRLS) algorithm. The summary of the LFXRLS algorithm is as follows

The weight update equation for the adaptive filters is given below

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \mu e(n) \mathbf{k}_i(n) \quad (4.12)$$

where, $i = 0, 1, \dots, P$ is the number of adaptive filters.

The individual Kalman gain vector is defined below

$$\mathbf{k}_i(n) = \frac{\mathbf{z}_i(n)}{\mathbf{s}_i(n) \mathbf{z}_i(n) + 1} \quad (4.13)$$

$$\mathbf{z}_i(n) = \lambda^{-1} \mathbf{Q}_i(n-1) \mathbf{s}_i^T(n) \quad (4.14)$$

and the inverse of the autocorrelation matrix

$$\mathbf{Q}_i(n) = \lambda^{-1} [\mathbf{Q}_i(n-1) - \mathbf{k}_i(n) \mathbf{z}_i(n)] \quad (4.15)$$

where $0 \leq \lambda < 1$ is the forgetting factor, which weights the recent data more heavily in order to accommodate nonstationary signals.

4.6 Fast LFXLMS Algorithm

The filter-bank implementation of the LFXLMS algorithm is shown in fig. 4.3. In such a scheme, the residual error sensed by the error microphone, is expressed by

$$e(n) = d(n) + b(n) * y(n) \quad (4.16)$$

In (4.16), $b(n)$ represents the impulse response of the secondary path transfer function and $y(n)$ is the output of the Legendre neural network, which is computed as

$$y(n) = \sum_{i=1}^P y_i(n) \quad (4.17)$$

where

$$y_i(n) = \mathbf{s}_i^T(n) \mathbf{w}_i(n) \quad (4.18)$$

and $i = 1, 2, \dots, P$, P is the order of function expansion. Using LFXLMS algorithm the weight update equations at time $n-1$ and n , respectively, are written as

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) - \mu e(n-1) \mathbf{s}_i'(n-1) \quad (4.19)$$

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) - \mu e(n) \mathbf{s}_i'(n) \quad (4.20)$$

where $\mathbf{s}_i'(n)$ is the input signal vector, $\mathbf{s}_i(n)$ filtered through the secondary path filter.

Output of the controller

The output of the controller at time $n-1$ and n , is written as

$$y_i(n-1) = \mathbf{s}_i(n-1) \mathbf{w}_i^T(n-1) \quad (4.21)$$

$$y_i(n) = \mathbf{s}_i(n) \mathbf{w}_i^T(n), \quad i = 1, 2, \dots, P \quad (4.22)$$

Inserting (4.19) into (4.22) yields

$$y_i(n) = \mathbf{s}_i(n) \mathbf{w}_i^T(n-1) - \mu e(n-1) \mathbf{s}_i(n) \mathbf{s}_i'^T(n-1) \quad (4.23)$$

From (4.21) and (4.23) we obtained

$$\begin{bmatrix} y_i(n-1) \\ y_i(n) \end{bmatrix} = \begin{bmatrix} \mathbf{s}_i(n-1) \\ \mathbf{s}_i(n) \end{bmatrix} \mathbf{w}_i^T(n-1) - \begin{bmatrix} 0 \\ \mu e(n-1) \mathbf{s}_i(n) \mathbf{s}_i'^T(n-1) \end{bmatrix} \quad (4.24)$$

Consider the first term on the right hand side of (4.24)

$$\begin{bmatrix} \mathbf{s}_i(n-1) \\ \mathbf{s}_i(n) \end{bmatrix} \mathbf{w}_i^T(n-1) = \begin{bmatrix} \mathbf{a}_{i,1} & \mathbf{a}_{i,2} \\ \mathbf{a}_{i,0} & \mathbf{a}_{i,1} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{i,0}^T(n-1) \\ \mathbf{w}_{i,1}^T(n-1) \end{bmatrix} \quad (4.25)$$

where

$$\begin{aligned} \mathbf{a}_{i,0} &= [s_i(n) \ s_i(n-2) \ \dots \ s_i(n-N+2)] \\ \mathbf{a}_{i,1} &= [s_i(n-1) \ s_i(n-3) \ \dots \ s_i(n-N+1)] \\ \mathbf{a}_{i,2} &= [s_i(n-2) \ s_i(n-4) \ \dots \ s_i(n-N)] \end{aligned} \quad (4.26)$$

and

$$\mathbf{w}_{i,j}(n) = [w_{i,j}(n) \ w_{i,j+2}(n) \ \dots \ w_{i,j+N-2}(n)]^T \quad \text{for } j=0, 1 \quad (4.27)$$

Adding and subtracting $\alpha_{i,1}\mathbf{w}_{i,1}(n-1)$ to the first row and $\alpha_{i,1}\mathbf{w}_{i,0}(n-1)$ to the second row on the right hand side of (4.25), the following expression is obtained

$$\begin{bmatrix} \mathbf{s}_i(n-1) \\ \mathbf{s}_i(n) \end{bmatrix} \mathbf{w}_i^T(n-1) = \begin{bmatrix} \alpha_{i,1}(\mathbf{w}_{i,0}(n-1) + \mathbf{w}_{i,1}(n-1)) & + (\alpha_{i,2} - \alpha_{i,1})\mathbf{w}_{i,1}(n-1) \\ \alpha_{i,1}(\mathbf{w}_{i,0}(n-1) + \mathbf{w}_{i,1}(n-1)) & + (\alpha_{i,1} - \alpha_{i,0})\mathbf{w}_{i,0}(n-1) \end{bmatrix} \quad (4.28)$$

As the $N/2$ element vector, $\alpha_{i,1}(\mathbf{w}_{i,0}(n-1) + \mathbf{w}_{i,1}(n-1))$, in (4.28) is common to both the rows, it requires only a one time computation. Thus, (4.28) requires $1.5NP$ multiplications for two time steps, which means $0.75NP$ multiplications on an average for each time step.

$$M_1 = 0.75NP$$

Therefore, this saves $0.25NP$ multiplications.

Now, consider the second term on the right-hand side of (4.23). Let

$$\psi_i(n) = \mathbf{s}_i(n)\mathbf{s}_i'^T(n-1) \quad (4.29)$$

Further $\psi_i(n)$ can be computed with less number of computations as follows:

$$\begin{aligned} \psi_i(n) = & \psi_i(n-2) + [s_i(n)s_i'(n-1) + s_i(n-1)s_i'(n-2)] \\ & - [s_i(n-N)s_i'(n-N-1) + s_i(n-N-1)s_i'(n-N-2)] \end{aligned} \quad (4.30)$$

Since the term $[s_i(n-N)s_i'(n-N-1) + s_i(n-N-1)s_i'(n-N-2)]$ in (4.30) has already been computed at time $n-N$, $\psi_i(n)$ requires only $2P$ multiplications for two time steps, where P is the total number of FIR adaptive filters.

The terms $\alpha_{i,2} - \alpha_{i,1}$ and $\alpha_{i,1} - \alpha_{i,0}$ in (4.28) require only one addition each, as all terms except the first term in these summations have already been computed at $n-2$ time step. Therefore, the number of multiplications required per time step, to compute (4.28), is given by $0.5P(N+2)$.

$$M_1 = 0.5P(N+2) \quad (4.31a)$$

In order to compute (4.30), we required $1.5P$ additions per sample. Therefore, the total number of additions required per sample to compute the output of the controller is equal to A_1

$$A_1 = 0.5P(N+5) \quad (4.31b)$$

Secondary Path Filtering

The terms $s'_i(n-1)$ and $s'_i(n)$ can be computed in a similar manner as done in case of $y_i(n-1)$ and $y_i(n)$ were computed in (4.24) and are shown below

$$s'_i(n-1) = \mathbf{s}_i(n-1)\mathbf{b}^T(n) \quad i = 1, 2, \dots, P \quad (4.32)$$

$$s'_i(n) = \mathbf{s}_i(n)\mathbf{b}^T(n) \quad (4.33)$$

$$\begin{bmatrix} s'_i(n-1) \\ s'_i(n) \end{bmatrix} = \begin{bmatrix} \beta_{i,1}(\mathbf{b}_0 + \mathbf{b}_1) + (\beta_{i,2} - \beta_{i,1})\mathbf{b}_1 \\ \beta_{i,1}(\mathbf{b}_0 + \mathbf{b}_1) - (\beta_{i,1} - \beta_{i,0})\mathbf{b}_0 \end{bmatrix} \quad (4.34)$$

where

$$\begin{aligned} \beta_{i,0} &= [s_i(n) \ s_i(n-2) \ \dots \ s_i(n-M+2)] \\ \beta_{i,1} &= [s_i(n-1) \ s_i(n-3) \ \dots \ s_i(n-M+1)] \\ \beta_{i,2} &= [s_i(n-2) \ s_i(n-4) \ \dots \ s_i(n-M)] \end{aligned} \quad (4.35)$$

where M is the order of the secondary path filter transfer function and

$$\mathbf{b}_j = [b_j \ b_{j+2} \ \dots \ b_{j+M-2}]^T \quad (4.36)$$

Equation (4.34) requires $1.5MP$ multiplications for two time steps, since the term $\beta_{i,1}(\mathbf{b}_0 + \mathbf{b}_1)$ has to be computed only once. Therefore, the number of multiplications required per time step is $0.75MP$,

$$M_2 = 0.75MP. \quad (4.37a)$$

The term $\mathbf{b}_0 + \mathbf{b}_1$ can be precomputed and stored. Hence, it requires no additions and the terms $\beta_{i,2} - \beta_{i,1}$ and $\beta_{i,1} - \beta_{i,0}$ requires one addition each. Therefore, the total number of additions required per sample is given by

$$A_2 = 0.5P(M+2) \quad (4.37b)$$

Weight Update

Substituting (4.33) into (4.32), we obtain

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n-1) - \mu e(n-1)s'_i(n-1) - \mu e(n)s'_i(n) \quad (4.38)$$

$$\begin{bmatrix} \mathbf{w}_{i,0}(n+1) \\ \mathbf{w}_{i,1}(n+1) \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{i,0}(n-1) \\ \mathbf{w}_{i,1}(n-1) \end{bmatrix} - \mu e(n-1) \begin{bmatrix} \gamma_{i,1}^T \\ \gamma_{i,2}^T \end{bmatrix} - \mu e(n) \begin{bmatrix} \gamma_{i,0}^T \\ \gamma_{i,1}^T \end{bmatrix} \quad (4.39)$$

where

$$\begin{aligned}
\gamma_{i,0} &= [s'_i(n) \ s'_i(n-2) \ \dots \ s'_i(n-N+2)] \\
\gamma_{i,1} &= [s'_i(n-1) \ s'_i(n-3) \ \dots \ s'_i(n-N+1)] \\
\gamma_{i,2} &= [s'_i(n-2) \ s'_i(n-4) \ \dots \ s'_i(n-N)]
\end{aligned} \tag{4.40}$$

and $\mathbf{w}_{i,0}(n)$ and $\mathbf{w}_{i,1}(n)$ are defined as in (4.27)

Rewriting (4.25) as

$$\begin{bmatrix} \mathbf{w}_{i,0}(n+1) \\ \mathbf{w}_{i,1}(n+1) \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{i,0}(n-1) \\ \mathbf{w}_{i,1}(n-1) \end{bmatrix} - \mu \begin{bmatrix} \gamma_{i,1}^T & \gamma_{i,0}^T \\ \gamma_{i,2}^T & \gamma_{i,1}^T \end{bmatrix} \begin{bmatrix} e(n-1) \\ e(n) \end{bmatrix} \tag{4.41}$$

Adding and subtracting $\mu \gamma_{i,1}^T e(n)$ to the first row and $\mu \gamma_{i,1}^T e(n-1)$ to the second row on the right-hand side of (4.41), the following equation is obtained

$$\begin{bmatrix} \mathbf{w}_{i,0}(n+1) \\ \mathbf{w}_{i,1}(n+1) \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{i,0}(n-1) \\ \mathbf{w}_{i,1}(n-1) \end{bmatrix} - \begin{bmatrix} \gamma_{i,1}^T \mu(e(n-1) + e(n)) - (\gamma_{i,1} - \gamma_{i,0})^T \mu e(n) \\ \gamma_{i,2}^T \mu(e(n-1) + e(n)) - (\gamma_{i,2} - \gamma_{i,1})^T \mu e(n-1) \end{bmatrix} \tag{4.42}$$

By following a similar analysis as we did for (4.34), the weight update (4.28) requires $0.75NP$ multiplications per iteration,

$$M_3 = 0.75NP \tag{4.43a}$$

The number of additions required to compute (4.42) is given by A_3

$$A_3 = 0.5P(2N+3) \tag{4.43b}$$

From (4.31a), (4.37a), and (4.43a), the number of multiplications per sample is given by

$M_1 + M_2 + M_3 = P(1.25N + 0.75M + 1)$, compared to the standard FFXLMS, which requires $(2P+1)(2N+M)$ multiplications. From (4.31b), (4.37b), and (4.43b) the total number of additions is given by $A_1 + A_2 + A_3 = 0.5P(3N + M + 10)$, whereas for the standard FFXLMS, this number is $(2P+1)(2N+M-2)$. Note that while comparing these algorithms, the multiplication due to step size (μ) has not been taken into account.

Table 4.1.

Computational Complexity Comparison

Algorithm		LFXLMS	LFELMS	FFXLMS	Fast LFXLMS
Number of multiplications required per sample for computation of	Controller Output	$N(P-1)$	$N(P-1)$	$N(2P+1)$	$0.5P(N+2)$
	Secondary path filtering	$M(P-1)$	M	$M(2P+1)$	$0.75MP$
	Weight update	$N(P-1)$	$N(P-1)$	$N(2P+1)$	$0.75NP$
	Total	$(2N+M)(P-1)$	$2N(P-1)+M$	$(2P+1)(2N+M)$	$P(1.25N+0.75M+1)$
Number of additions required per sample for computation of	Controller Output	$(N-1)(P-1)$	$(N-1)(P-1)$	$(N-1)(2P+1)$	$0.5P(N+5)$
	Secondary path filtering	$(M-1)(P-1)$	$M-1$	$(M-1)(2P+1)$	$0.5P(M+2)$
	Weight update	$N(P-1)$	$N(P-1)$	$N(2P+1)$	$0.5P(2N+3)$
	Total	$(2N+M-2)(P-1)$	$(2N-1)(P-1)+M-1$	$(2P+1)(2N+M-2)$	$0.5P(3N+M+10)$

4.7 Simulation and Results

Extensive simulation work has been done for various nonlinear ANC and some selected results are presented to validate the proposed algorithms. The performance of the proposed LFXLMS algorithm, LFELMS algorithm and LFXRLS algorithm are compared with FLANN based algorithm (FFXLMS algorithm). A number of nonlinear ANC with linear secondary path and nonlinear secondary path are tested. Simulation result of fast-LFXLMS is also compared with LFXLMS algorithm. Mean square error (MSE) in dB defined by

$$\text{MSE(dB)} = 10 \log_{10} \{E(e^2(n))\} \quad (4.44)$$

is plotted for each simulation. In each of the experiments, hundred independent trials are conducted and the average MSE(dB) is plotted to obtain smoother convergence characteristics.

4.7.1 Experiment I

In the first experiment a nonlinear ANC with the primary path transfer function defined below is considered [64]

$$A(z) = z^{-5} - 0.3z^{-6} + 0.2z^{-7} \quad (4.45)$$

The secondary path considered is a non-minimum-phase filter with transfer function

$$B(z) = z^{-2} + 1.5z^{-3} - z^{-4} \quad (4.46)$$

Secondary path is assumed to be perfectly estimated i.e. $\hat{B}(z) = B(z)$

The reference noise is the logistic chaotic noise generated by the following equation [11]

$$x(n+1) = \lambda x(n)[1 - x(n)] \quad (4.47)$$

where $\lambda = 4$ and $x(0) = 0.9$ are used. This noise process is then normalized to have unit signal power. MSE(dB) for proposed LFXLMS, LFELMS and LFXRLS algorithms are plotted and compared with that of FFXLMS algorithm. The step size for LFXLMS, LFELMS, FFXLMS, LFXRLS algorithms are 0.0004, 0.0003, 0.0004, 0.005 respectively and forgetting factor for LFXRLS algorithm considered is 0.99. MSE(dB) plots for all the algorithms are shown in the fig. 4.8. The steady state MSE(dB) obtained by LFXLMS, LFELMS, LFXRLS and FFXLMS algorithm are -30dB, -29 dB, -32 dB and -26 dB respectively. The proposed algorithms results in lower steady state MSE(dB) compared to FFXLMS algorithm which indicates LNN based algorithms perform better than FFXLMS algorithm.

In order to verify tracking capability of the developed algorithms the primary path transfer function and/or secondary path transfer function are varied after the algorithms entered into convergence region. In the first test primary path is changed (the undesired noise at the cancellation point is changed from $d(n)$ to $-d(n)$) at 3000th iteration. The MSE(dB) plot is shown in fig. 4.9, which confirms that all the algorithms successfully converged even after variation in primary path. In the second test the secondary path transfer function is changed from $B(z) = z^{-2} + 1.5z^{-3} - z^{-4}$ to

$B(z) = 0.9z^{-2} + 1.1z^{-3} - 1.1z^{-4}$ at 3000th iteration and the MSE(dB) plots obtained are shown in the fig. 4.10. In the third test both the primary path and secondary path are varied at 3000th iteration and the MSE(dB) plots are shown in the fig. 4.11. From all the three tests it can be concluded that the proposed algorithms are able to track variation in the primary and secondary paths.

4.7.2 Experiment II

In order to analyze the performance of the proposed algorithms in case of nonlinear active noise canceller, nonlinear primary path as well as nonlinear secondary path is considered here. In this experiment the nonlinear primary path is defined by the following primary to desired signal relationship [62]

$$d(n) = x(n) + 0.8x(n-1) + 0.3x(n-2) + 0.4x(n-3) - 0.8x(n)x(n-1) + 0.9x(n)x(n-2) + 0.7x(n)x(n-3) \quad (4.48)$$

Similarly nonlinear secondary path considered has the following input to output relationship

$$\hat{d}(n) = y(n) + 0.35y(n-1) + 0.09y(n-2) - 0.5y(n)y(n-1) + 0.4y(n)y(n-2) \quad (4.49)$$

Reference signal is considered to be white noise. MSE(dB) for the proposed LFXLMS and LFELMS algorithms are obtained using virtual secondary path and compared with that of FFXLMS algorithm. The step size for LFXLMS, LFELMS and FFXLMS algorithms are 0.0002, 0.0001 and 0.0002 respectively. The steady state MSE(dB) obtained by LFXLMS, LFELMS and LFXRLS algorithm are -14dB, -12 dB and -12 dB respectively. The proposed LFXLMS algorithm outperformed FFXLMS algorithm in terms of steady state MSE(dB). But the vital advantage of the proposed algorithms is their low computational complexity.

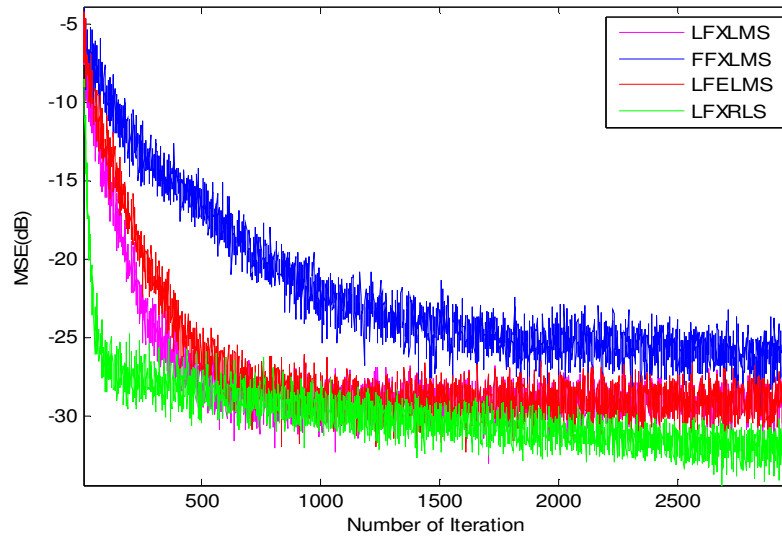


Fig. 4.8 MSE(dB) plot for LFXLMS, FFXLMS, LFELMS and LFXRLS algorithm.

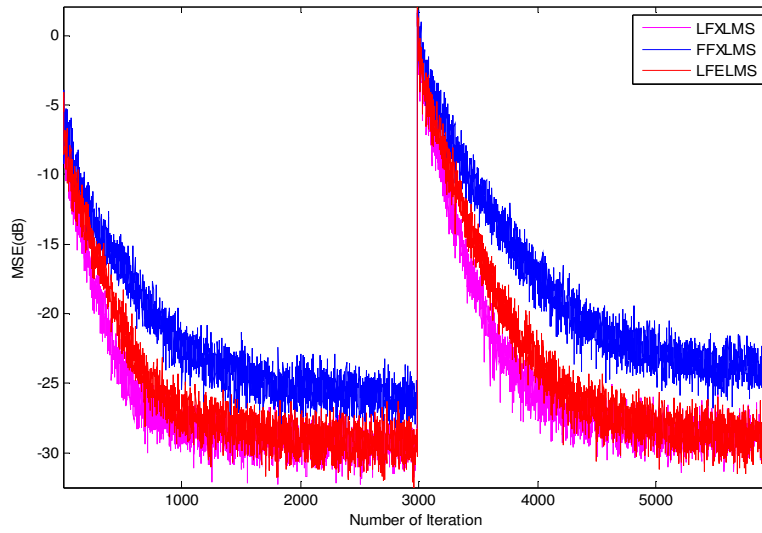


Fig. 4.9 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm when primary path is changed at 3000th iteration.

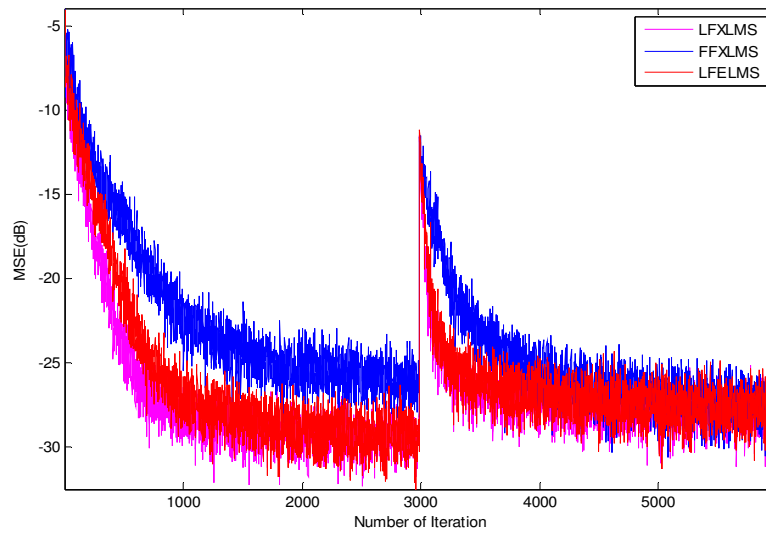


Fig. 4.10 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm when secondary path is changed at 3000th iteration.

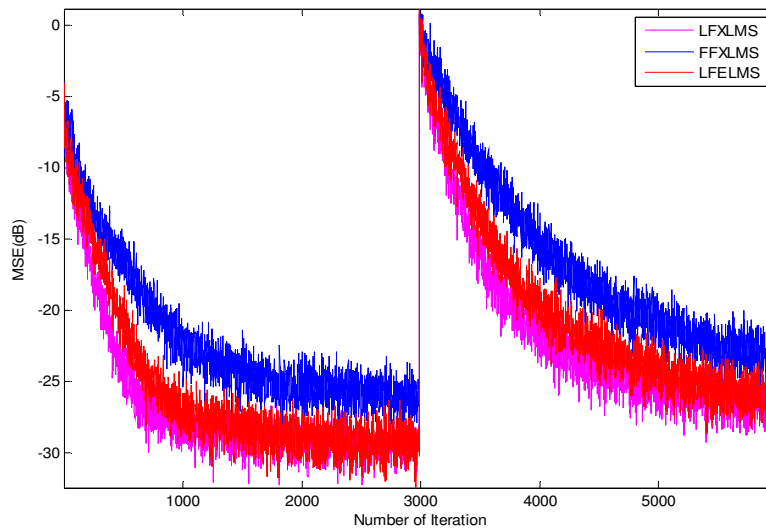


Fig. 4.11 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm when both primary path and secondary path are changed at 3000th iteration.

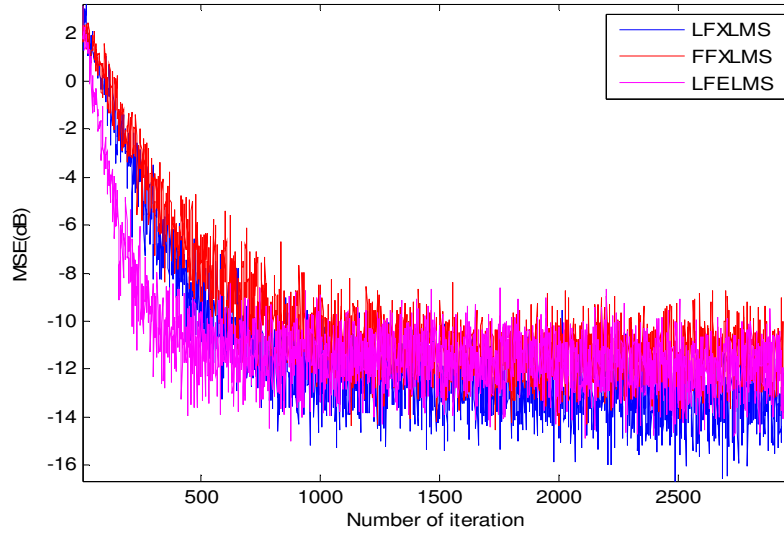


Fig. 4.12 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm.

4.7.3 Experiment III

Another experiment is conducted on the nonlinear active noise canceller. In this experiment the nonlinear primary path is defined by the following input to output relation [62]

$$d(n) = x(n-5) + 0.8x(n-6) + 0.3x(n-7) + 0.4x(n-8) + 0.2x(n-5)x(n-6) - 0.3x(n-5)x(n-7) + 0.4x(n-5)x(n-8) \quad (4.50)$$

The nonlinear secondary path considered is in cascade form. It consist of cascading of three blocks, Linear (l_1)-Nonlinear (N)-Linear (l_2) (LNL) shown in fig. 4.13.

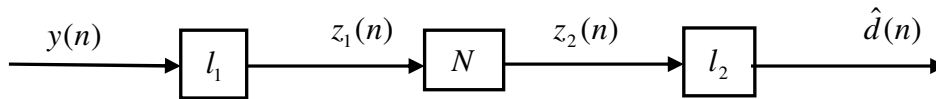


Fig. 4.13 Block diagram of LNL nonlinear secondary path model.

The blocks l_1, N, l_2 are defined as follows

$$l_1 = 1 - 0.6z^{-1} + 0.05z^{-2} \quad (4.51)$$

$$N(z_1) = 3.3 \tanh[0.3z_1(n)] \quad (4.52)$$

$$l_2 = 1 + 0.2z^{-1} + 0.05z^{-2} \quad (4.53)$$

The reference noise considered here is white noise. The step size for LFXLMS, LFELMS and FFXLMS algorithms are 0.0003, 0.002 and 0.0003 respectively. Proceeding in the same manner as in [62] the MSE(dB) obtained by LFXLMS, LFELMS and FFXLMS algorithm are plotted in the fig. 4.14, which suggest equivalent steady state MSE(dB) for all the three algorithms. But here it should be noted that the proposed algorithms have lower computational complexity requirement.

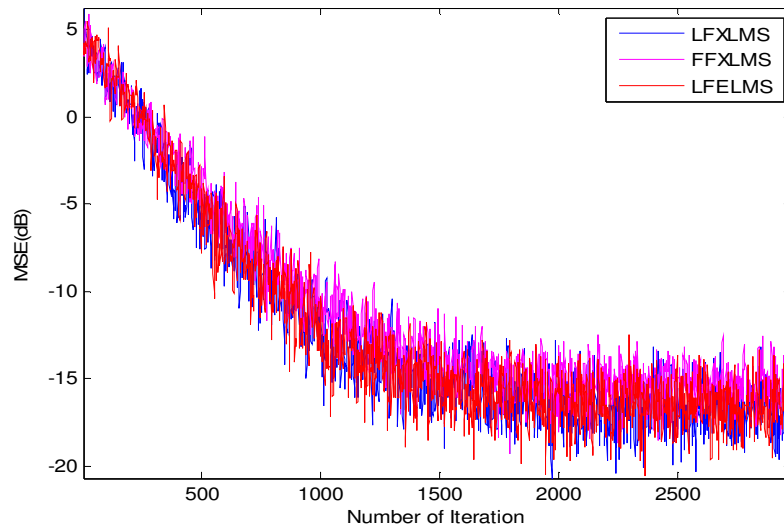


Fig. 4.14 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm.

4.7.4 Experiment IV

The primary path, secondary path and input signal are same as same that of experiment I. But adaptive algorithm for weight update is fast LFXLMS algorithm and LFXLMS algorithm. Step size for both the algorithms is 0.01. MSE(dB) plot for the algorithms are shown in fig.4.15. Both the algorithms yield identical results but real advantage of fast LFXLMS algorithm is lower computational complexity.

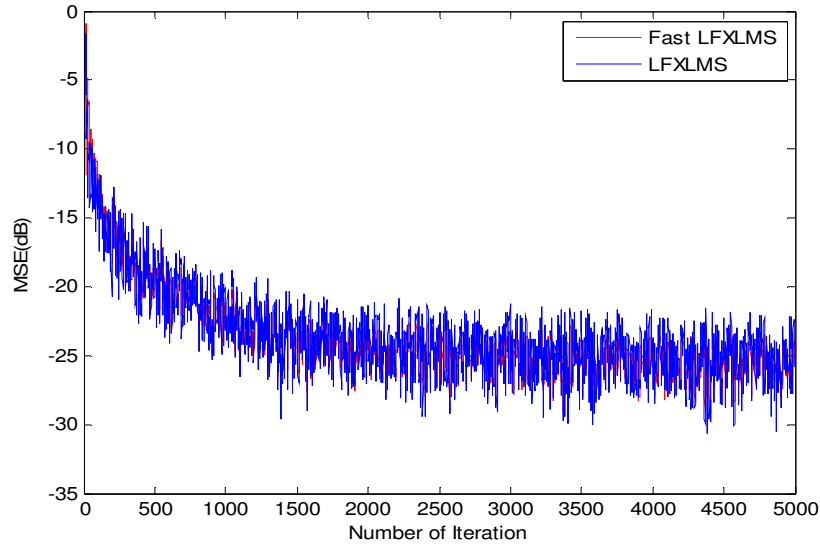


Fig. 4.15 MSE(dB) plot for LFXLMS algorithm and Fast LFXLMS algorithm.

4.7.5 Experiment on Real Time Signal

In order to assess the performance of the developed algorithms in a real time environment a few experiments are conducted on real time signals as reference noise. Two reference signals considered are Buccaneer Jet cockpit noise (used in previous chapter) and white noise. White Noise is acquired by sampling high-quality analog noise generator (Wandel & Goltermann) [79]. It exhibits equal energy per Hz. bandwidth.

4.7.6 Experiment V (Real Time Signal)

In this experiment two reference signals picked up from real world environment are considered. Fig.4.16 and fig.4.17 shows MSE(dB) plots of LFXLMS, FFXLMS and LFELMS algorithms for Buccaneer jet cockpit noise and white noise respectively. The primary path and secondary path are identical to the experiment-1. Step size(Buccaneer jet cockpit noise) for LFXLMS is 0.0005, FFXLMS is 0.0005 and LFELMS is 0.0004. The step size (white noise) for LFXLMS is 0.0004, FFXLMS is 0.0005 and LFELMS is 0.0003. The steady state MSE(dB) obtained by LFXLMS, FFXLMS and LFELMS

algorithm in case of jet cockpit noise are -27dB, -25dB and -27dB and in case of white noise are -23dB, -22dB and -22dB respectively.

4.7.7 Experiment VI (Real Time Signal)

In order to verify the tracking capability of the developed algorithms the primary path transfer function is varied after the algorithms entered into convergence region. The primary path is changed (the undesired noise at the cancellation point is changed from $d(n)$ to $-d(n)$) at 3000th iteration. Fig.4.18, fig.4.19 shows MSE(dB) plots of LFXLMS and FFXLMS algorithms for Buccaneer jet cockpit noise and white noise respectively. Step size (Buccaneer jet cockpit noise) for LFXLMS is 0.0005 and FFXLMS is 0.0005. The step size (white noise) for LFXLMS is 0.0004 and FFXLMS is 0.0005. The steady state MSE(dB) obtained by LFXLMS and FFXLMS algorithms confirms that all the algorithms successfully converged after variation in primary path.

4.7.8 Experiment VII (Real Time Signal)

Experiment-VI is conducted once again but this time secondary path rather than primary path is varied after the ANC entered into steady state region. The secondary path transfer function is changed from $B(z) = z^{-2} + 1.5z^{-3} - z^{-4}$ to $B(z) = 0.9z^{-2} + 1.1z^{-3} - 1.1z^{-4}$ at 3000th iteration and the MSE(dB) plots obtained are shown in the fig.4.20 and fig.4.21 for Buccaneer jet cockpit noise and white noise respectively. From the above two tests it can be concluded that the proposed algorithms are able to track variations in the primary and secondary paths.

4.7.9 Experiment VIII (Real Time Signal)

Performances of the proposed algorithms are assessed in nonlinear environment. Here both the primary and secondary paths are considered nonlinear and their transfer functions are identical to experiment-II. Fig.4.22 and fig.4.23 shows MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithms for Buccaneer jet cockpit noise and white noise respectively. The steady state MSE(dB) obtained for LFXLMS, FFXLMS and LFELMS algorithms in case of Jet Cockpit Noise are -9dB, -4dB, -7dB and in case of white noise are -10dB, -5dB and -9dB.

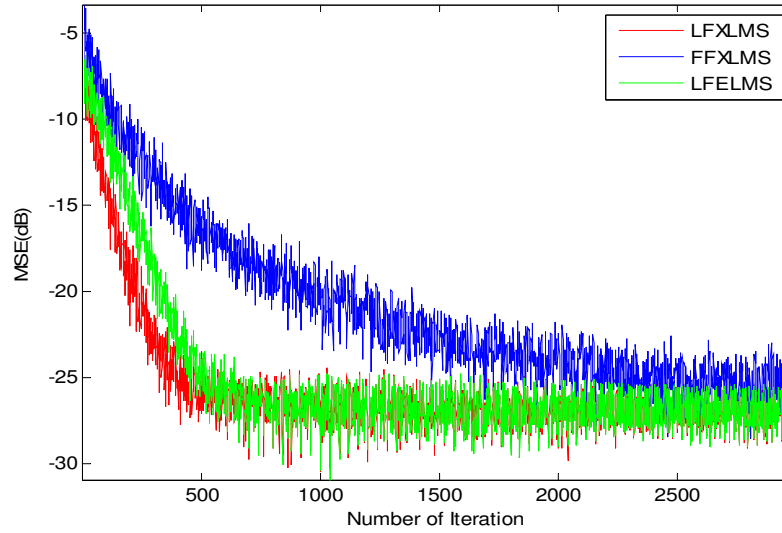


Fig. 4.16 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm(Jet Cockpit Noise).

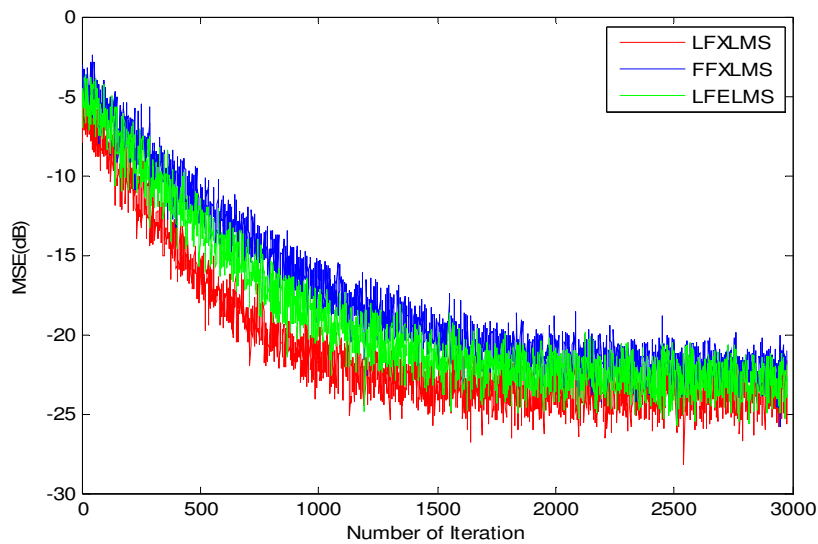


Fig. 4.17 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm(White Noise).

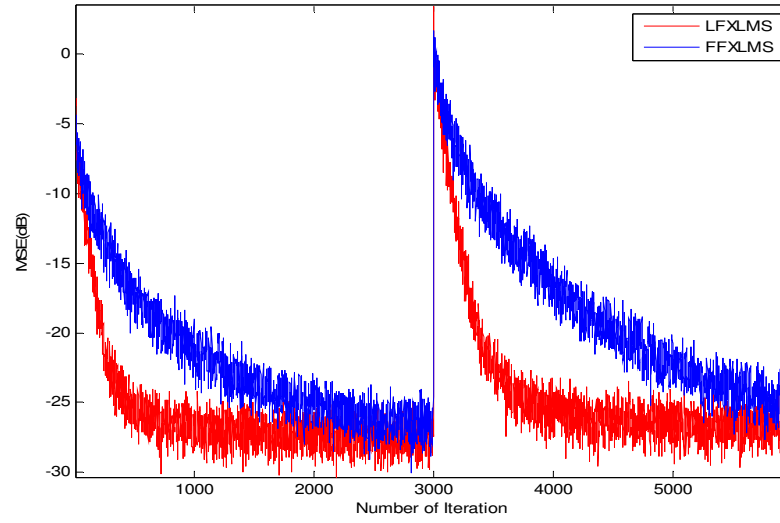


Fig. 4.18 MSE(dB) plot for LFXLMS and FFXLMS algorithm when primary path is changed at 3000th iteration(Jet Cockpit Noise).

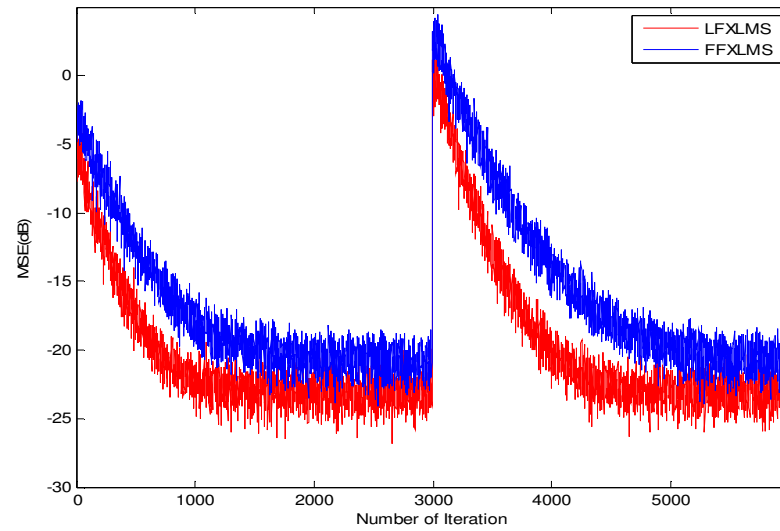


Fig. 4.19 MSE(dB) plot for LFXLMS and FFXLMS algorithm when primary path is changed at 3000th iteration(White Noise).

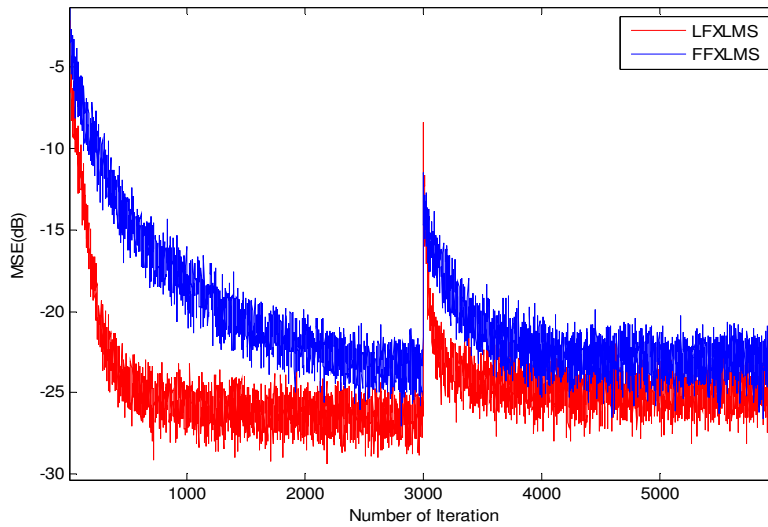


Fig. 4.20 MSE(dB) plot for LFXLMS and FFXLMS algorithm when secondary path is changed at 3000th iteration(Jet Cockpit Noise).

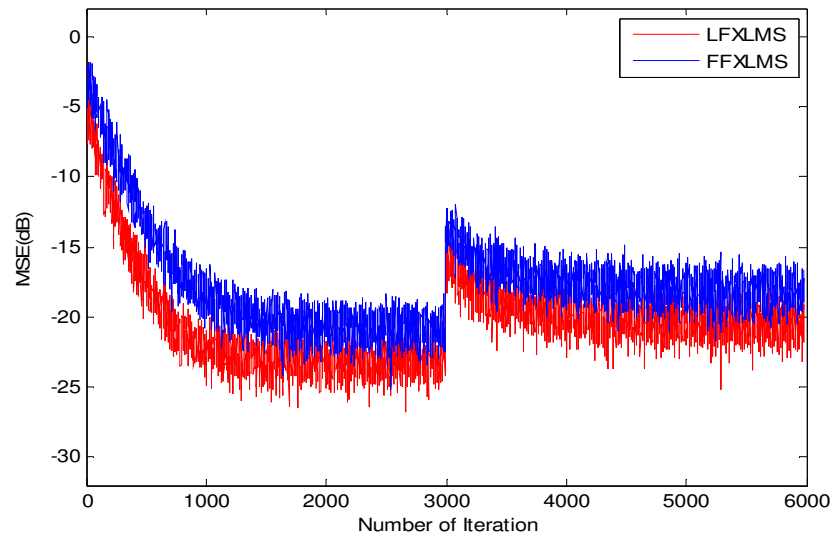


Fig. 4.21 MSE(dB) plot for LFXLMS and FFXLMS algorithm when secondary path is changed at 3000th iteration(White Noise).

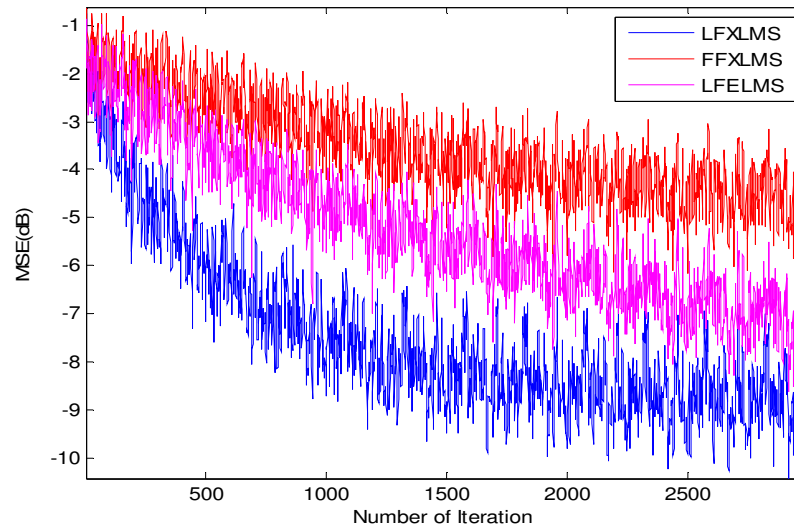


Fig. 4.22 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm(Jet Cockpit Noise).

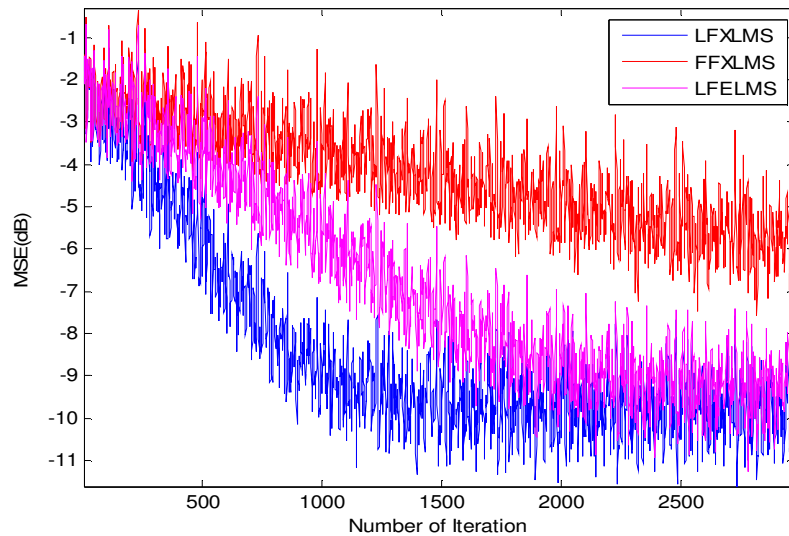


Fig. 4.23 MSE(dB) plot for LFXLMS, FFXLMS and LFELMS algorithm(White Noise).

4.8 Summary

This chapter proposes a computationally efficient reduced structure Legendre neural network for nonlinear active noise cancellation. Update algorithms Legendre FXLMS (LFXLMS) and Legendre FELMS (LFELMS) for the proposed network are derived. RLS algorithm is also employed to develop Legendre FXRLS (LFXRLS) algorithm to obtain faster convergence but at the cost of increase in computational complexity. A fast version of LFXLMS algorithm called fast LFXLMS algorithm is also developed which reduces computational complexity by almost 25%. Extensive simulation are conducted for various reference noises and the MSE(dB) plots are obtained. Experiments are also conducted considering real time reference signals such as Buccaneer jet cockpit noise and white noise. The experimental results presented here demonstrates the superior performance of the proposed algorithms in terms of MSE(dB) and computational complexity compared to standard FFXLMS algorithm.

CHAPTER 5

Frequency-Domain Approach to Multichannel Nonlinear Active Noise Control

5.1 Background

Active noise control has been a field of growing interest over the past few decades. Traditionally ANCs are realized by adaptive filtering in time domain. Adaptive filtering in frequency domain is an attractive alternative to time domain adaptive filtering. The theory for frequency domain adaptive filtering is well developed and literature is thick. Frequency domain filters have primarily two advantages compared to time domain implementation. The first advantage is the potentially large scale savings in the computational complexity. The Fast Fourier Transform (FFT) is an efficient implementation of the Discrete Fourier Transform (DFT) which provides this savings. A second advantage is that DFT generate signals that are approximately uncorrelated (orthogonal). As a result a time varying step size can be used for each adaptive weight, thereby allowing faster convergence. Another secondary advantage of frequency domain adaptation is more accurate estimation of gradient due to the averaging of samples in a whole data block.

Several researchers have implemented the ANC in frequency domain using different variations of the FXLMS algorithm. Q. Shen and A. S. Spanias [73], G. A. Clark, S. K. Mitra, and S. R. Parkar [74] proposed block implementation of the FXLMS algorithm, both in time and frequency domain, which is exact implementation of FXLMS algorithm. Sen M. Kuo, Mansour Tahernezehadi and Li Ji [75], M. R. Asharif, T. Takebayashi, T. Chugo and K. Murano[76], Reichard and Swanson[77], D. P. Das, G. Panda and S. M. Kuo[78] proposed different methods for frequency domain implementation of active noise cancellation.

In this chapter, a simple and computationally efficient frequency domain algorithm for multichannel ANC is proposed. In addition, normalized LMS [3], [7]

algorithm is employed to facilitate variable step size. This new algorithm is termed as multichannel frequency domain block filtered-x normalized LMS (FBFXNLMS) algorithm. E. Wan [14] proposed adjoint LMS algorithm which led to the development FELMS algorithm which is an efficient alternative to FXLMS algorithm with an advantage of reduction in computational complexity for multiple input ANC. A frequency domain implementation of FELMS algorithm has been proposed and is termed as frequency domain block filtered-e least mean square (FBFELMS) algorithm. Legendre neural network for nonlinear ANC, developed in the previous chapter is also implemented in frequency domain using Fast Fourier Transform (FFT). The developed algorithms are termed as frequency domain block Legendre filtered-x least mean square (FBLFXLMS) and frequency domain block Legendre filtered-e least mean square (FBLFELMS) algorithm.

5.2 The Filtered-x Least Mean Square (FXLMS) Algorithm

While developing the time domain block FXLMS algorithm, the FXLMS algorithm is first outlined. The FXLMS is the most common algorithm applied in both feedforward and feedback ANC due to its ease in implementation [2]. The basic ANC system is shown in fig 5.1 where the path from the noise source to the cancellation point is defined as primary path and it has a transfer function, $A(z)$. ANC also has secondary

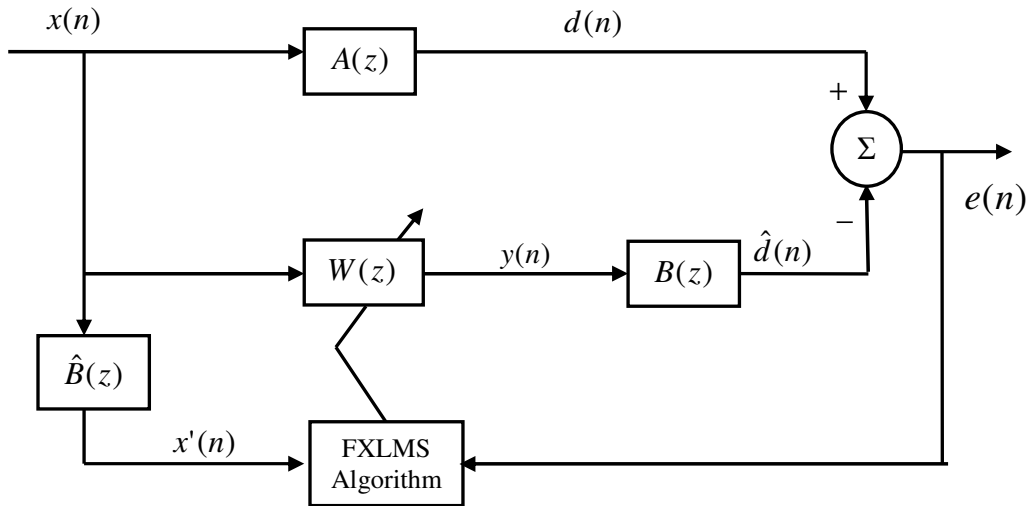


Fig. 5.1 Block diagram of the basic active noise control system.

path which is defined as the path leading from the adaptive filter output to error sensor that measures the residual noise and it has a transfer function, $B(z)$. Most available ANC algorithms including FXLMS, require online or offline identification of secondary path.

In an ANC system the secondary path transfer function, $B(z)$ follows the adaptive filter. Therefore, to ensure convergence, the conventional LMS algorithm is to be suitably modified. The most appropriate modification is by placing an estimate of this secondary path transfer function, $\hat{B}(z)$ in the reference signal path to the weight update of the LMS algorithm. Hence the algorithm is referred to as filtered-x LMS (FXLMS) algorithm. Referring to fig. 5.1, the residual noise signal at n^{th} time instant is expressed as

$$e(n) = d(n) - \hat{d}(n) \quad (5.1)$$

$$= d(n) - \mathbf{y}(n) * \hat{\mathbf{b}}(n) \quad (5.2)$$

$$= d(n) - \mathbf{w}(n) * \mathbf{x}(n) * \hat{\mathbf{b}}(n) \quad (5.3)$$

where $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]$.

$d(n)$ = noise to be cancelled at the canceling point,

$\hat{d}(n)$ = output of the ANC,

$\mathbf{b}(n)$ = the impulse response of the secondary path transfer function,

$\hat{\mathbf{b}}(n)$ = estimate of the impulse response of the secondary path transfer function,

and $*$ denotes linear convolution operation.

$\mathbf{w}(n)$ = estimate of the weight vector at n^{th} instant,

$$= [w_0(n) \ w_1(n) \ \dots \ w_{N-1}(n)]^T$$

The weight update equation in FXLMS algorithm is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{c}(n) \quad (5.4)$$

where μ = convergence coefficient.

$$\mathbf{c}(n) = \mathbf{x}'(n) * e(n) \quad (5.5)$$

$$\mathbf{x}'(n) = \mathbf{x}(n) * \mathbf{b}(n) \quad (5.6)$$

The FXLMS algorithm is thus described by (5.1) through (5.6), which involves three convolution operations to compute $\mathbf{x}'(n)$, $\mathbf{y}(n)$, and $\hat{d}(n)$. Out of these three linear

convolution operations, two are performed in actual implementation to evaluate $\mathbf{x}'(n)$, and $\mathbf{y}(n)$. But in case of ANC system the third convolution is actually not computed as the signal $\hat{d}(n)$ corresponds to output of the speaker and is used as an antinoise signal. In the next section the time-domain block filtered-x least mean square (BFXLMS) algorithm is derived in detail.

5.3 The Time Domain Block Filtered-x LMS (BFXLMS) Algorithm

In block filtering instead of computing the ANC output sample by sample a block of output is computed simultaneously and this is possible by using overlap save or overlap add method. Out of these two methods the overlap-save method has been chosen as it is computationally more efficient. Now using overlap-save method the FXLMS algorithm is implemented by redefining (5.5) as follows

$$\mathbf{c}(n) = \mathbf{x}'(n) * \mathbf{e}(n)$$

$$\text{where } \mathbf{e}(n) = [e(n-N+1) \dots e(n-1) e(n)]$$

Computation of $\mathbf{c}(n)$ actually involve cross correlation, but writing $\mathbf{e}(n)$ in time increasing order the cross correlation operation is converted to convolution operation. In block FXLMS the computation of the three linear convolution operations (for obtaining $\mathbf{x}'(n)$, $\mathbf{y}(n)$, and $\mathbf{c}(n)$) can be implemented in a simple manner as:

$$\mathbf{y}(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] \left\{ \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{x}(n+N) \end{bmatrix} * \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{w}(n) \right\} \quad (5.7)$$

$$\mathbf{x}'(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] \left\{ \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{x}(n+N) \end{bmatrix} * \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \hat{\mathbf{b}}(n) \right\} \quad (5.8)$$

$$\mathbf{c}(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] \left\{ \begin{bmatrix} \mathbf{x}'(n) \\ \mathbf{x}'(n+N) \end{bmatrix} * \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}(n+N) \right\} \quad (5.9)$$

where the matrix \mathbf{I}_N is an $N \times N$ identity matrix and the matrix \mathbf{O}_N is an $N \times N$ matrix with all zero elements and $\mathbf{x}(n+N) = [x(n+N) \ x(n+N-1) \dots x(n+1)]$.

5.4 Frequency Domain Block Filtered-x LMS (FBFXLMS)

Algorithm

Fast Fourier Transform can be used to efficiently compute the associated linear convolutions in BFXLMS algorithm. It is shown in the previous section that the BFXLMS algorithm essentially consists of three linear convolutions defined in (5.7) to (5.9). FFT based implementation of all the three linear convolutions can be done by defining F_{2N} and F_{2N}^{-1} as the 2N-point FFT and IFFT operators respectively. The linear convolution in (5.7) may be implemented using the 2N point FFT and IFFT as

$$\mathbf{X}(n+N) = F_{2N} \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{x}(n+N) \end{bmatrix} \quad (5.10)$$

$$\mathbf{W}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{w}(n) \quad (5.11)$$

$$\mathbf{y}(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{X}(n+N) \otimes \mathbf{W}(n)] \quad (5.12)$$

where \otimes denotes point-by-point multiplication. Similarly, (5.8) may be implemented using the 2N point FFT and IFFT as

$$\mathbf{B}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{b}(n) \quad (5.13)$$

$$\mathbf{x}'(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{X}(n+N) \otimes \mathbf{B}(n)] \quad (5.14)$$

The FFT-based implementation of (5.9) can be written as

$$\mathbf{X}'(n+N) = F_{2N} \begin{bmatrix} \mathbf{x}'(n) \\ \mathbf{x}'(n+N) \end{bmatrix} \quad (5.15)$$

$$\mathbf{E}(n+N) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}(n+N) \quad (5.16)$$

$$\mathbf{c}(n+N) = [\mathbf{O}_N \quad \mathbf{T}_N] F_{2N}^{-1} [\mathbf{X}'(n+N) \otimes \mathbf{E}(n+N)] \quad (5.17)$$

The weight update equation of FBFXLMS algorithm becomes

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{c}(n+N) \quad (5.18)$$

Here weights are updated in time domain where as all the linear convolution operations required for executing the algorithm is done in frequency domain.

5.5 Frequency Domain Block Filtered-x Normalized LMS (FBFXNLMS) Algorithm

The conventional linear adaptive filters based on the normalized LMS (NLMS) algorithm obtain rapid convergence for highly correlated signals, by improving a large eigenvalue spread. With similar reasoning, the proposed approach relies upon the NLMS algorithm in order to solve the slow convergence problem. Accordingly FXLMS algorithm is modified to yield filtered-x normalized least mean square (FXNLMS) algorithm. The weight update equation for FXNLMS algorithm is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\mathbf{x}(n)^T \mathbf{x}(n) + \varepsilon} \mathbf{x}'(n)e(n) \quad (5.19)$$

where ε is a small constant. Incorporating block implementation and subsequently employing FFTs and IFFTs to reduce computational load the weight update equation (5.19) becomes

$$\mathbf{w}(n+N) = \mathbf{w}(n) + \frac{\mu}{\mathbf{x}(n+N)^T \mathbf{x}(n+N) + \varepsilon} \mathbf{c}(n+N) \quad (5.20)$$

This equation represents the frequency domain block filtered-x normalized LMS (FBFXNLMS) algorithm.

5.5.1 The Reduced Structure FBFXNLMS algorithm

Further reduction in computational complexity can be achieved by carefully observing the block diagram of fig.5.2 and discarding some steps. To complete the weight update process the reference signal is multiplied with the estimated secondary path in frequency domain. After multiplication the resulting signal undergoes two transformations; first from frequency domain to time domain and then again back to frequency domain. The steps followed are described as follows. (i) From frequency domain to time domain by a $2N$ point IFFT (ii) The first N samples are deleted (iii) Overlapping with previous N samples is done (iv) The signal is again transformed to frequency domain by a $2N$ point FFT. Thus process is to transform a frequency domain signal to time domain and again transforming the time domain signal to frequency domain. If these two transformations are removed, the above mentioned four steps become redundant but still the performance is preserved. So the resulting signal after reference signal is multiplied with the estimated secondary path in frequency domain is

directly used for weight update. By removing these steps, an approximate version is obtained [78]. In this way one FFT and one IFFT computations can be saved. The complete algorithm is shown in table 5.1. The block diagram of ANC using the FBFXLMS algorithm is shown in fig. 5.3.

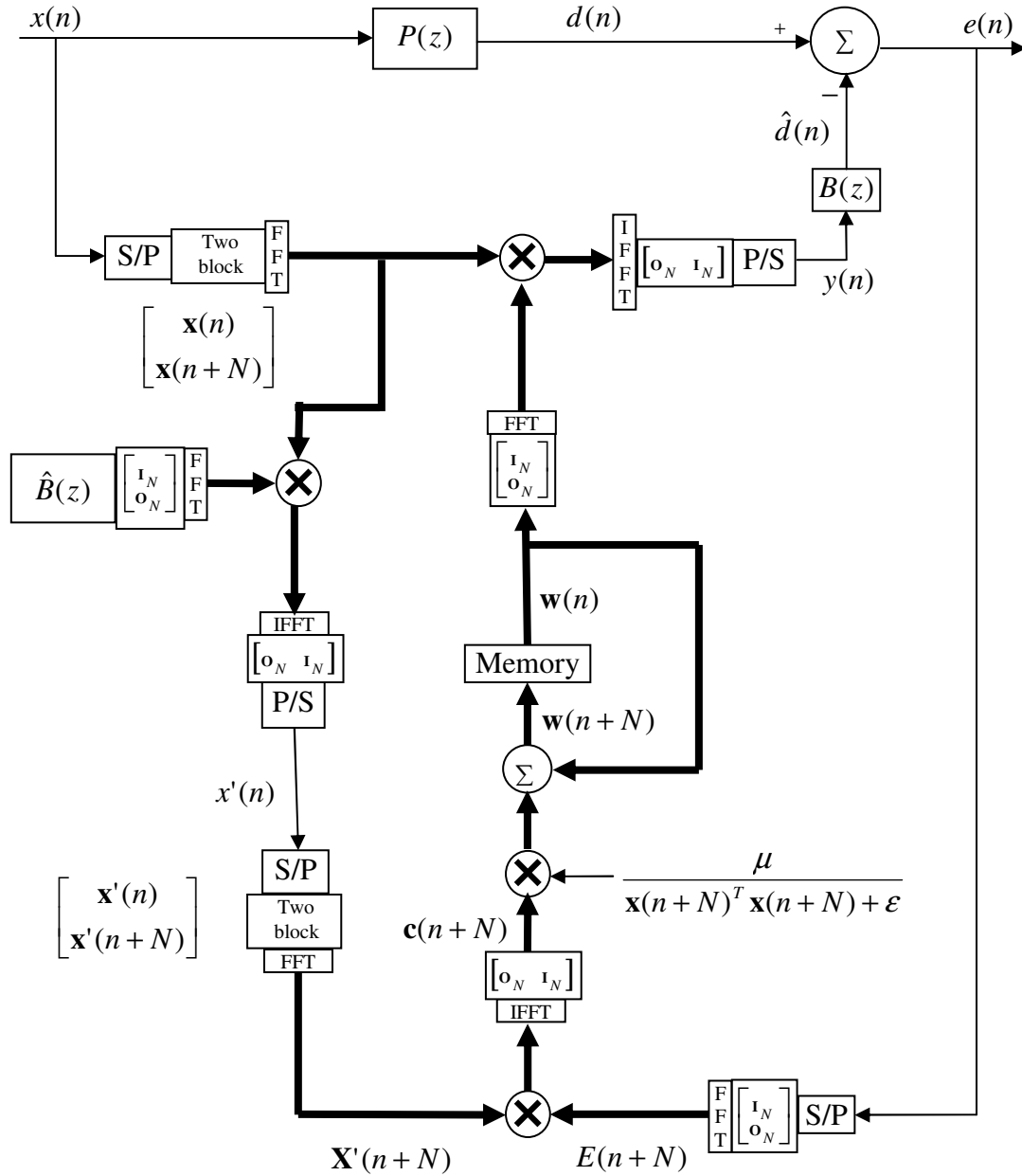


Fig.5.2 Block diagram of ANC using FBFXLMS algorithm.

5.6 Frequency Domain Block Filtered-e LMS (FBFELMS) Algorithm

Filtered-e least mean square (FELMS) algorithm is widely used as an alternative to FXLMS algorithm in multiple input multiple output (MIMO) ANC which reduces computational complexity and memory requirements. Adjoint LMS algorithm for ANC was developed by E. A. Wan [14] which is a FELMS algorithm where error signal rather than input signal is filtered through adjoint of secondary path filter. Incorporating block processing and using FFT and IFFT, FELMS algorithm can be implemented in frequency domain as follows.

$$\mathbf{X}(n+N) = F_{2N} \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{x}(n+N) \end{bmatrix} \quad (5.21)$$

$$\mathbf{W}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{w}(n) \quad (5.22)$$

$$\mathbf{y}(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{X}(n+N) \otimes \mathbf{W}(n)] \quad (5.23)$$

where \otimes denotes point-by-point multiplication. Similarly filtering of error block through estimated secondary path may be implemented using the 2N point FFT as follows

$$\mathbf{B}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{b}_r(n) \quad (5.24)$$

$$\mathbf{E}(n+N) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}(n+N) \quad (5.25)$$

$$\mathbf{E}'(n+N) = [\mathbf{E}(n+N) \otimes \mathbf{B}(n)] \quad (5.26)$$

where $\mathbf{b}_r(n)$ is obtained by flipping $\mathbf{b}(n)$.

The FFT-based implementation of weight adaptation can be written as

$$\mathbf{c}(n+N) = [\mathbf{O}_N \quad \mathbf{T}_N] F_{2N}^{-1} [\mathbf{X}(n+N) \otimes \mathbf{E}'(n+N)] \quad (5.27)$$

$$\mathbf{w}(n+N) = \mathbf{w}(n) + \mu \mathbf{c}(n+N) \quad (5.28)$$

The complete algorithm is shown in table 5.2.

Table 5.1

FBFXNLMS algorithm

Controller Output:

$$\mathbf{X}(n+N) = F_{2N} \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{x}(n+N) \end{bmatrix}$$

$$\mathbf{W}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{w}(n)$$

$$\mathbf{y}(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{X}(n+N) \otimes \mathbf{W}(n)]$$

Filtering through secondary path:

$$\mathbf{B}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{b}(n)$$

$$\mathbf{x}'(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{X}(n+N) \otimes \mathbf{B}(n)]$$

Weight update:

$$\mathbf{X}'(n+N) = F_{2N} \begin{bmatrix} \mathbf{x}'(n) \\ \mathbf{x}'(n+N) \end{bmatrix}$$

$$\mathbf{E}(n+N) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}(n+N)$$

$$\mathbf{c}(n+N) = [\mathbf{O}_N \quad \mathbf{T}_N] F_{2N}^{-1} [\mathbf{X}'(n+N) \otimes \mathbf{E}(n+N)]$$

$$\mathbf{w}(n+N) = \mathbf{w}(n) + \frac{\mu}{\mathbf{x}(n+N)^T \mathbf{x}(n+N) + \varepsilon} \mathbf{c}(n+N)$$

Table 5.2

FBFELMS algorithm

Controller Output:

$$\mathbf{X}(n+N) = F_{2N} \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{x}(n+N) \end{bmatrix}$$

$$\mathbf{W}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{w}(n)$$

$$\mathbf{y}(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{X}(n+N) \otimes \mathbf{W}(n)]$$

Filtering through secondary path:

$$\mathbf{B}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{b}_r(n)$$

$$\mathbf{E}(n+N) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}(n+N)$$

$$\mathbf{E}'(n+N) = [\mathbf{E}(n+N) \otimes \mathbf{B}(n)]$$

Weight update:

$$\mathbf{c}(n+N) = [\mathbf{O}_N \quad \mathbf{T}_N] F_{2N}^{-1} [\mathbf{X}(n+N) \otimes \mathbf{E}'(n+N)]$$

$$\mathbf{w}(n+N) = \mathbf{w}(n) + \mu \mathbf{c}(n+N)$$

5.7 Multichannel FBFXNLMS Algorithm

Noise control in a large dimension duct or enclosure requires a multiple channel ANC systems. Multiple channel ANC engage several secondary loudspeakers to control noise at multiple error microphone locations. It is assumed that in a multiple channel ANC, L number of reference microphones, P number of secondary loudspeakers and Q numbers of error microphones are employed. So in total LP numbers of adaptive filters are present and their impulse responses are represented as $\mathbf{w}_{lp}(n)$ and PQ number of secondary paths is represented as $\mathbf{b}_{pq}(n)$. Applying multiple error LMS algorithm, proposed by Elliott [1], [2], multiple channel ANC problem can be solved by applying FBFXNLMS to all possible single channel paths in the multiple channel system. The weight update equation can be written as

$$\mathbf{w}_{lp}(n+1) = \mathbf{w}_{lp}(n) + \frac{\mu}{\mathbf{x}_l(n+N)^T \mathbf{x}_l(n+N) + \varepsilon} \sum_{q=1}^Q \mathbf{c}_{lq}(n) \quad (5.29)$$

For $1 < l < L$ and $1 < p < P$ and

$$\mathbf{c}_{lq}(n+N) = [\mathbf{O}_N \quad \mathbf{T}_N] F_{2N}^{-1} [\mathbf{X}'_l(n+N) \otimes \mathbf{E}_q(n+N)] \quad (5.30)$$

$$\mathbf{X}'_l(n+N) = \mathbf{X}_l(n+N) \otimes \mathbf{B}_{pq} \quad (5.31)$$

$$\mathbf{B}_{pq} = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{b}_{pq} \quad (5.32)$$

$$\mathbf{E}_q(n+N) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}_q(n+N) \quad (5.33)$$

where \mathbf{b}_{pq} is the impulse response of the secondary path extending from p^{th} loudspeaker and q^{th} error microphone.

5.7.1 Computational Complexity

In case of FXNLMS algorithm to obtain N samples of controller outputs, LPN^2 multiplications and $LPN(N-1)$ additions are required. For filtering N samples of reference signal through the secondary path of length N, $LPQN^2$ multiplications and $LPQN(N-1)$ additions are required. For weight update, $LP(Q+1)N^2$ multiplications and $LP(Q+1)N^2$ additions are required. Therefore the total number of multiplications required is

$2LP(Q+1)N^2$ and the total additions required is $NLP(Q+1)(2N-1)$. For FBFXNLMS algorithm, single channel ANC using overlap save method, the N -point FBFXNLMS algorithm involves the computation of (i) six $2N$ -point FFTs, (ii) three $2N$ point complex multiplications and (iii) N number of weight updates. For real-valued input data, total number of real multiplications is $12N\log_2 N + 24N$ and the real additions is $24N\log_2(N) + 13N$.

In case of multichannel ANC the number of $2N$ point FFT/IFFT required for (i) input signal transform is L , (ii) adaptive filter output signal transform is P , (iii) adaptive filter transform is LP , (iv) secondary path transfer function transform is PQ , (v) error signal transform is Q , (vi) transform of product of filtered input signal and error is LP . So total number of FFT is $L+P+2LP+PQ+Q$. Each FFT requires $2N\log_2(N)$ real multiplications and $4N\log_2(N)$ real additions. Also LP, LPQ, LPQ number of $2N$ point frequency domain complex multiplications are required for computing adaptive filter output, filtered input signal, weight update respectively. Each $2N$ point complex multiplication involves $8N$ real multiplications and $4N$ real additions. Also the number of real additions required for weight update is $LPN+2LPQN$. So total real multiplications required is $(L+P+2LP+PQ+Q)2N\log_2(N) + (LP+2LPQ)8N$ and real additions required is $(L+P+2LP+PQ+Q)4N\log_2(N) + (LP+2LPQ)4N + LPN + 2LPQN$. Computational complexity for multichannel FXNLMS and multichannel FBFXNLMS algorithms with $L=2, P=2, Q=2$ are tabulated in table 5.3 and plotted in fig. 5.5.

Table 5.3.

Computational Complexity per sample

N	Number of multiplication		Number of addition	
	FXNLMS	FBFXNLMS	FXNLMS	FBFXNLMS
32	768	340	756	532
64	1536	376	1524	604
128	3072	412	3060	676
256	6144	448	6132	748
512	12288	484	12276	820
1024	24576	520	24564	892

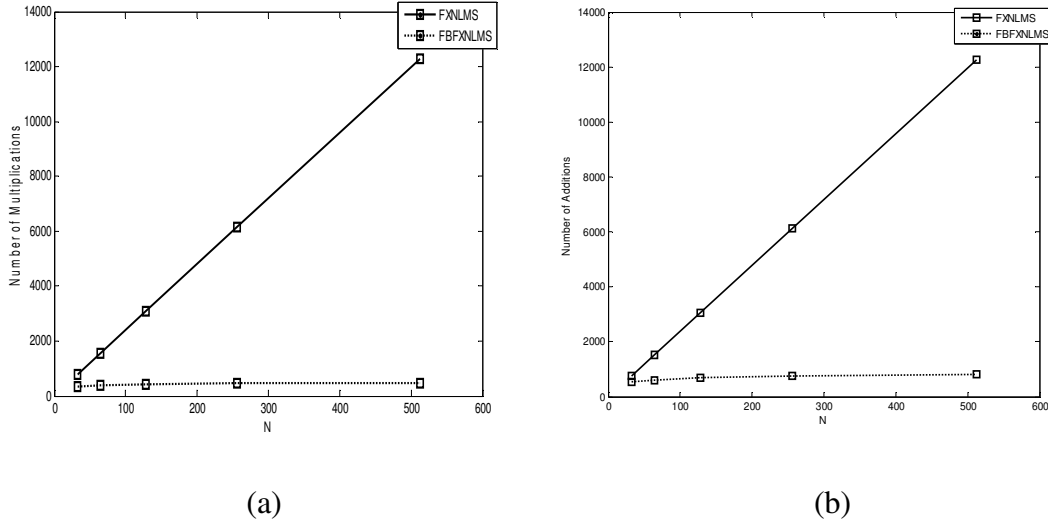


Fig.5.5 Comparison of Computational Complexity (a) Multiplications (b) Additions.
From table 5.3 and fig 5.5 saving in computational requirement can be observed.

5.8 Frequency Domain Implementation of Legendre Neural Network for Nonlinear ANC

The Block diagram of the nonlinear ANC using Legendre neural network (LNN) as the controller, developed in the previous chapter, is shown in fig.5.6. Here the reference noise vector $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]$ is transformed into an output vector $\mathbf{s}(n)$ given by $\mathbf{s}(n) = L(\mathbf{x}(n))$. The nonlinear function $L(\mathbf{x}(n))$ represents a set of the orthogonal basis functions, implemented in the ‘‘Legendre expansion’’ block. Here the N-dimensional input pattern $\mathbf{x}(n)$ is enhanced to an NP-dimensional enhanced pattern $\mathbf{s}(n)$ given by

$$\begin{aligned} \mathbf{s}(n) &= [\mathbf{s}_0(n) \ \mathbf{s}_1(n) \ \dots \ \mathbf{s}_{P-1}(n)] \\ &= [L_0(\mathbf{x}(n)) \ L_1(\mathbf{x}(n)) \ \dots \ L_{P-1}(\mathbf{x}(n))] \end{aligned} \quad (5.34)$$

where

$$\mathbf{s}_0(n) = L_0(\mathbf{x}(n)) = [1 \ 1 \ \dots \ N \text{ number of } 1]$$

$$\mathbf{s}_1(n) = L_1(\mathbf{x}(n)) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]$$

$$\mathbf{s}_2(n) = L_2(\mathbf{x}(n)) = \left[\frac{1}{2}(3x(n)^2 - 1) \ \frac{1}{2}(3x(n-1)^2 - 1) \ \dots \ \frac{1}{2}(3x(n-N+1)^2 - 1) \right]$$

.....

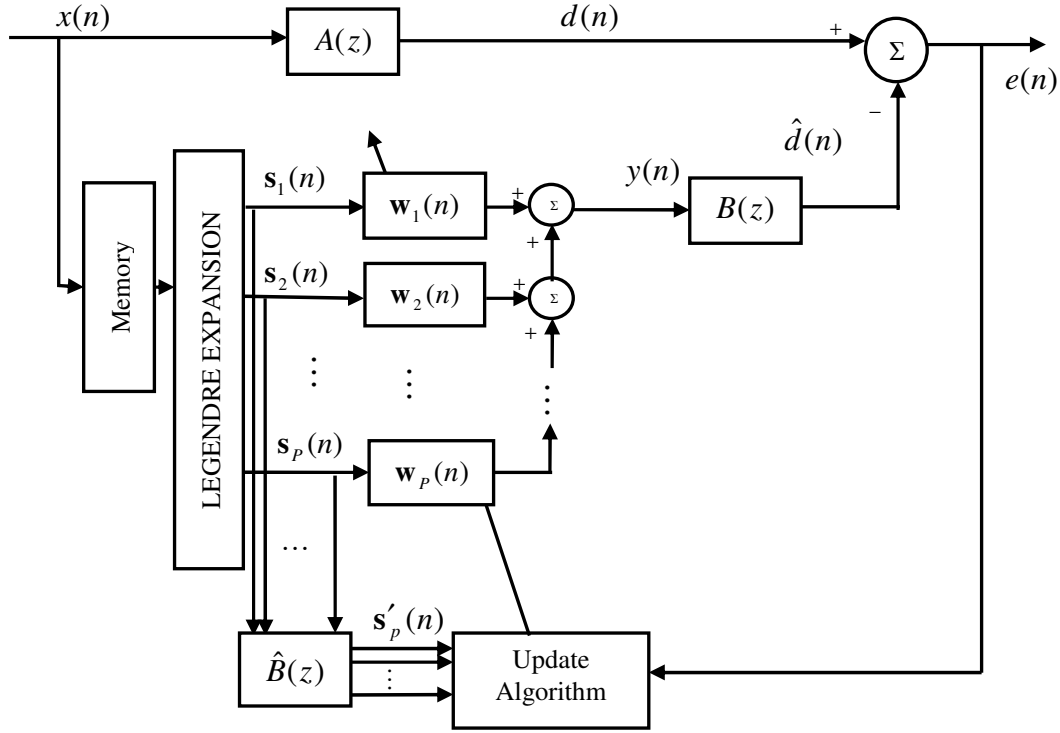


Fig.5.6 Legendre Neural Network for Nonlinear ANC.

For reduced structure LNN the $s_0(n)$ is discarded. Employing filter bank implementation output of LNN at time n is

$$y(n) = \sum_{i=1}^P y_i(n) = \sum_{i=1}^P s_i(n) \mathbf{w}_i(n) \quad (5.35)$$

where $\mathbf{w}_i(n)$ is the weight vector of i^{th} adaptive filter at time n . Estimated desired signal $\hat{d}(n)$ is obtained by filtering LNN output through secondary path $B(z)$. Error at

time n is defined as $e(n) = d(n) - \hat{d}(n)$. Considering a cost function of $\xi(n) = \frac{1}{2} E[e^2(n)]$

and using the FXLMS algorithm the all the weight vectors can be updated separately as

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \mu e(n) \mathbf{s}_i'(n) \quad \text{for } i=1, 2, \dots, P \quad (5.36)$$

where $\mathbf{s}_i'(n)$ is the input signal $\mathbf{s}_i(n)$ filtered through the estimated secondary path and μ is the step size which control convergence and stability.

Exploiting the properties of FFT the above algorithm can be implemented in frequency domain with block filtering which reduces the computational burden of the algorithm. In this case all the convolutions and correlations are done in frequency domain. This algorithm is termed as frequency domain block Legendre FXLMS (FBLFXLMS) algorithm. Defining F_{2N} and F_{2N}^{-1} as the $2N$ -point FFT and IFFT the linear convolutions are implemented as follows

$$\mathbf{S}_i(n+N) = F_{2N} \begin{bmatrix} \mathbf{s}_i(n) \\ \mathbf{s}_i(n+N) \end{bmatrix} \quad (5.37)$$

$$\mathbf{W}_i(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{w}_i(n) \quad (5.38)$$

$$\mathbf{y}_i(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}_i(n+N) \otimes \mathbf{W}_i(n)] \quad (5.39)$$

where \otimes denotes frequency-by-frequency bin multiply or in more general way point-by-point multiplication of two vectors. The \mathbf{I}_N is an $N \times N$ identity matrix, the matrix \mathbf{O}_N is an $N \times N$ matrix with all zero elements. Similarly filtering of reference signal through the estimated secondary path can be done in frequency domain as follows

$$\mathbf{B}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{b}(n) \quad (5.40)$$

$$\mathbf{s}_i'(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}_i(n+N) \otimes \mathbf{B}(n)] \quad (5.41)$$

Calculation of $\mathbf{c}_i(n+N)$, which are required for weight update are done in the following steps.

$$\mathbf{S}_i'(n+N) = F_{2N} \begin{bmatrix} \mathbf{s}_i'(n) \\ \mathbf{s}_i'(n+N) \end{bmatrix} \quad (5.42)$$

$$\mathbf{E}(n+N) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}(n+N) \quad (5.43)$$

$$\mathbf{c}_i(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}_i'(n+N) \otimes \mathbf{E}(n+N)] \quad (5.44)$$

The weight vectors are updated as follows

$$\mathbf{w}_i(n+N) = \mathbf{w}_i(n) + \mu \mathbf{c}_i(n+N) \quad (5.45)$$

The block diagram of ANC using Legendre neural network in frequency domain and updated by FBLFXLMS algorithm is shown in the fig. 5.7 and fig. 5.8. The complete algorithm is briefed in table 5.4

In the reduced structure FBFXNLMS algorithm two FFTs are saved by removing two FFTs in the calculation of $\mathbf{X}'(n+N)$. Here also two FFTs can be saved by removing two FFTs in the calculation of $\mathbf{S}_i'(n+N)$. For P^{th} order Legendre expansion in total $2P$ number of FFTs can be saved.

5.9 Frequency Domain Block Legendre FELMS (FBLFELMS) Algorithm.

In an ANC using Legendre neural network each of the expanded reference signal vectors $\mathbf{s}_i(n) (i=1,2,\dots, P)$ is to be filtered through the estimated secondary path. Number of filtering can be reduced by applying FELMS algorithm where only error signal is to be filtered through the estimated secondary path. Taking advantage of this fact implementation of ANC using Legendre neural network has been tried in frequency domain by employing FELMS algorithm. The resulting algorithm is named as frequency domain block Legendre FELMS algorithm (FBLFELMS) algorithm.

Referring to the previous section the steps of FBLFELMS algorithm is described as follows. Defining F_{2N} and F_{2N}^{-1} as the $2N$ -point FFT and IFFT, the linear convolutions are implemented as given below

$$\mathbf{S}_i(n+N) = F_{2N} \begin{bmatrix} \mathbf{s}_i(n) \\ \mathbf{s}_i(n+N) \end{bmatrix} \quad (5.46)$$

$$\mathbf{W}_i(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{w}_i(n) \quad (5.47)$$

$$\mathbf{y}_i(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}_i(n+N) \otimes \mathbf{W}_i(n)] \quad (5.48)$$

where \otimes denotes frequency-by-frequency bin multiply or in more general way point-by-point multiplication of two vectors. The \mathbf{I}_N is an $N \times N$ identity matrix, the matrix \mathbf{O}_N is an $N \times N$ matrix with all zero elements. Similarly another convolution can be done as

$$\mathbf{B}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{b}_r(n) \quad (5.49)$$

$$\mathbf{E}(n+N) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}(n+N) \quad (5.50)$$

$$\mathbf{E}'(n+N) = [\mathbf{E}(n+N) \otimes \mathbf{B}(n)] \quad (5.51)$$

and cross correlation can be done as

$$\mathbf{S}'_i(n+N) = F_{2N} \begin{bmatrix} \mathbf{s}'_i(n) \\ \mathbf{s}'_i(n+N) \end{bmatrix} \quad (5.52)$$

$$\mathbf{c}_i(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}'_i(n+N) \otimes \mathbf{E}'(n+N)] \quad (5.53)$$

So the weights are updated as follows

$$\mathbf{w}_i(n+N) = \mathbf{w}_i(n) + 2\mu \mathbf{c}_i(n+N) \quad (5.54)$$

The block diagram of ANC using Legendre neural network in frequency domain and updated by FBLFELMS algorithm is shown in the fig. 5.7 and fig. 5.9. The complete algorithm is briefed in table 5.5.

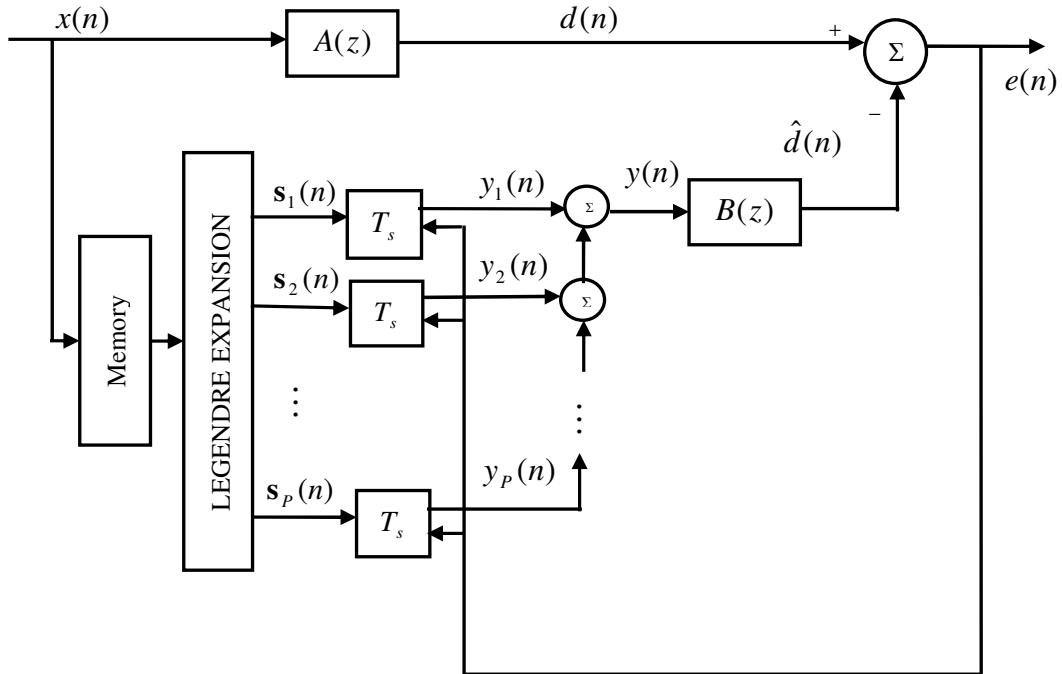


Fig.5.7 Frequency domain implementation of Legendre Neural Network for ANC.

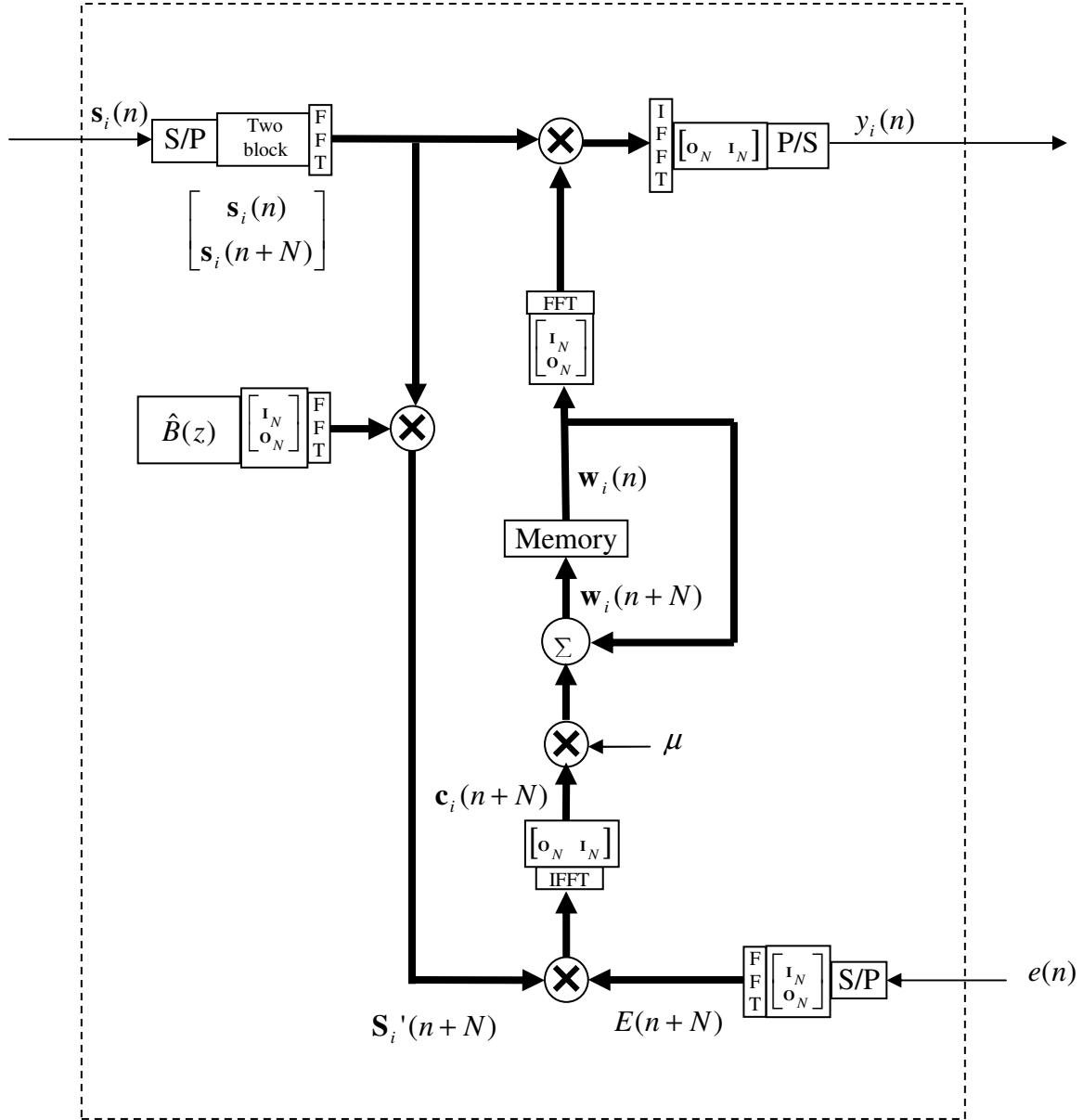
Fig.5.8 Details of T_s block of fig. 5.7 (for FBLFXLMS Algorithm).

Table 5.4

FBLFXLMS algorithm

Controller Output:

$$\mathbf{S}_i(n+N) = F_{2N} \begin{bmatrix} \mathbf{s}_i(n) \\ \mathbf{s}_i(n+N) \end{bmatrix}$$

$$\mathbf{W}_i(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{w}_i(n)$$

$$\mathbf{y}_i(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}_i(n+N) \otimes \mathbf{W}_i(n)]$$

Filtering through secondary path:

$$\mathbf{B}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{b}(n)$$

$$\mathbf{s}_i'(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}_i(n+N) \otimes \mathbf{B}(n)]$$

Weight update:

$$\mathbf{S}_i'(n+N) = F_{2N} \begin{bmatrix} \mathbf{s}_i'(n) \\ \mathbf{s}_i'(n+N) \end{bmatrix}$$

$$\mathbf{E}(n+N) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}(n+N)$$

$$\mathbf{c}_i(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}_i'(n+N) \otimes \mathbf{E}(n+N)]$$

$$\mathbf{w}_i(n+N) = \mathbf{w}_i(n) + \mu \mathbf{c}_i(n+N)$$

Table 5.5

FBLFELMS algorithm

Controller Output:

$$\mathbf{S}_i(n+N) = F_{2N} \begin{bmatrix} \mathbf{s}_i(n) \\ \mathbf{s}_i(n+N) \end{bmatrix}$$

$$\mathbf{W}_i(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{w}_i(n)$$

$$\mathbf{y}_i(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}_i(n+N) \otimes \mathbf{W}_i(n)]$$

Filtering through secondary path:

$$\mathbf{B}(n) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{b}_r(n)$$

$$\mathbf{E}(n+N) = F_{2N} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{O}_N \end{bmatrix} \mathbf{e}(n+N)$$

$$\mathbf{E}'(n+N) = [\mathbf{E}(n+N) \otimes \mathbf{B}(n)]$$

Weight update:

$$\mathbf{S}'_i(n+N) = F_{2N} \begin{bmatrix} \mathbf{s}'_i(n) \\ \mathbf{s}'_i(n+N) \end{bmatrix}$$

$$\mathbf{c}_i(n+N) = [\mathbf{O}_N \quad \mathbf{I}_N] F_{2N}^{-1} [\mathbf{S}'_i(n+N) \otimes \mathbf{E}'(n+N)]$$

$$\mathbf{w}_i(n+N) = \mathbf{w}_i(n) + 2\mu \mathbf{c}_i(n+N)$$

5.10 Simulation and Results

To analyze the performance of the proposed frequency domain multichannel ANC algorithms extensive simulation experiments studies have been carried out. Results of some of the experiments are shown here. In all the experiments mean square error (MSE) in dB defined by

$$\text{MSE(dB)} = 10 \log_{10} (E(e^2(n))) \quad (5.55)$$

is obtained through simulation. MSE in dB plotted is average of twenty independent runs of the experiments to get a smoother curve.

5.10.1 Experiment I

In this experiment, a multichannel ANC with one reference microphone, two loudspeakers and two error microphones are considered ($1 \times 2 \times 2$). Memory size N is chosen to be 10. In this experiment the two linear primary path transfer functions considered are described by [62]

$$A_{11}(z) = z^{-5} - 0.3z^{-6} + 0.2z^{-7}$$

$$A_{12}(z) = z^{-5} - 0.4z^{-6} + 0.1z^{-7}$$

and the four secondary path transfer functions considered are minimum-phase FIR models described below

$$B_{11}(z) = z^{-2} + 0.5z^{-3}$$

$$B_{12}(z) = z^{-2} + 0.6z^{-3}$$

$$B_{21}(z) = 0.9z^{-2} + 0.4z^{-3}$$

$$B_{22}(z) = 0.9z^{-2} + 0.3z^{-3}$$

Reference noise is taken as white noise. MSE(dB) is obtained for the proposed frequency domain multichannel FBFXNLMS algorithm taking step size $\mu=0.09$. For comparison with corresponding time domain algorithm, MSE(dB) for FXNLMS algorithm with $\mu=0.05$ is also obtained. Simulation results are plotted in fig.5.10 and fig.5.11 respectively. Steady state MSE(dB) achieved by FBFXNLMS algorithm and FXNLMS algorithm is found to be same -27dB. From the results, it is evident that the proposed algorithm offers identical performance as the time domain FXNLMS algorithm

for multichannel ANC but the real advantage of the proposed algorithm is large saving in computational complexity.

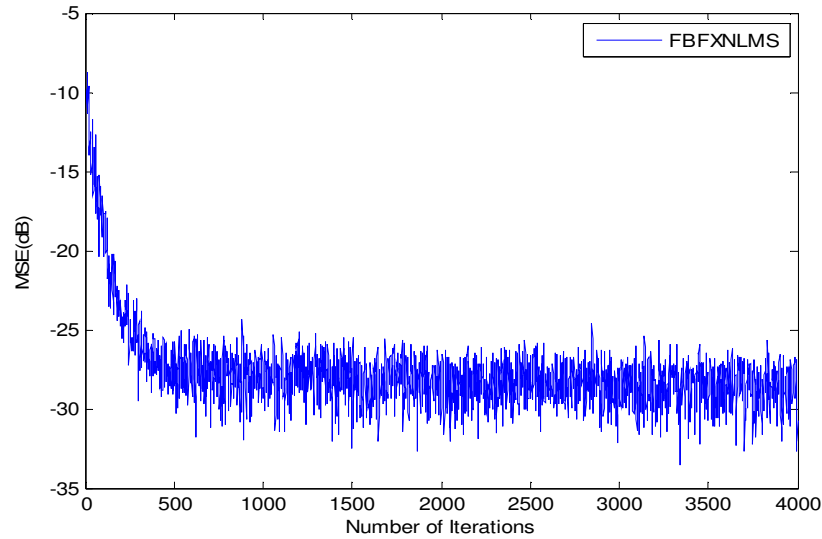


Fig. 5.10 MSE(dB) plot of FBFXNLMS algorithm for multichannel ANC.

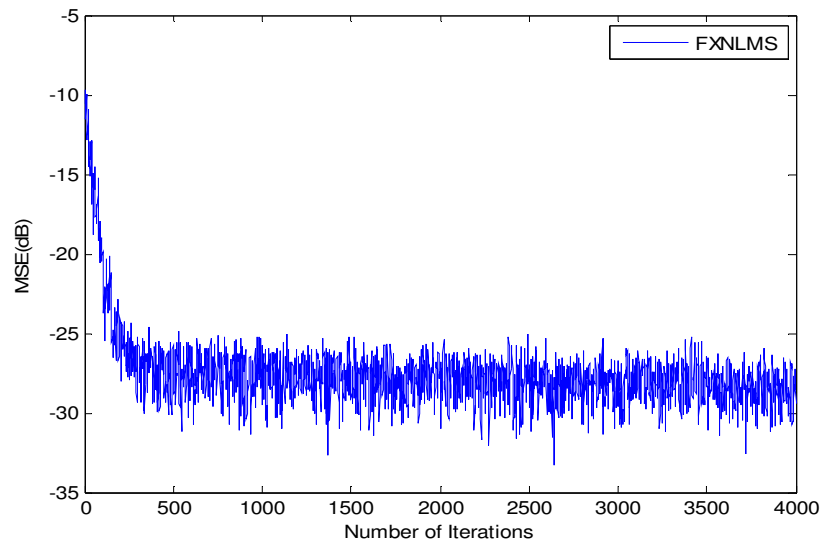


Fig. 5.11 MSE(dB) plot of FXNLMS algorithm for multichannel ANC.

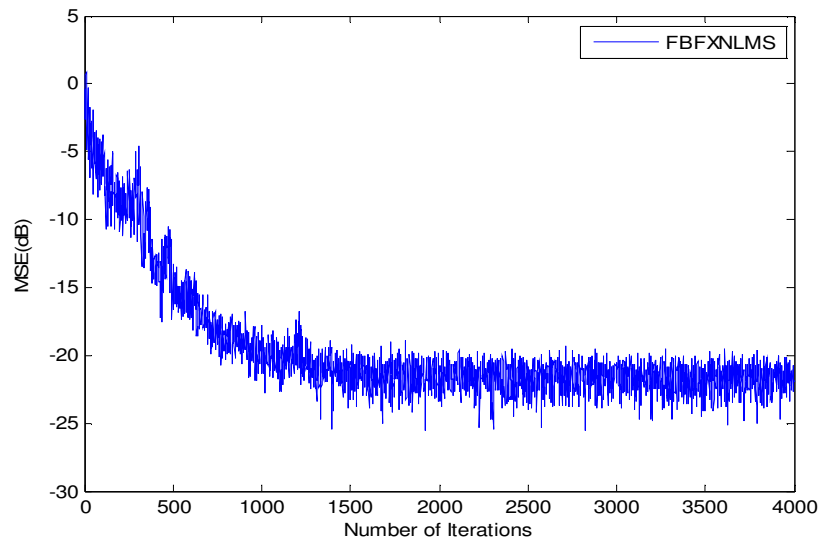


Fig. 5.12 MSE(dB) plot of FBFXNLMS algorithm for multichannel ANC.

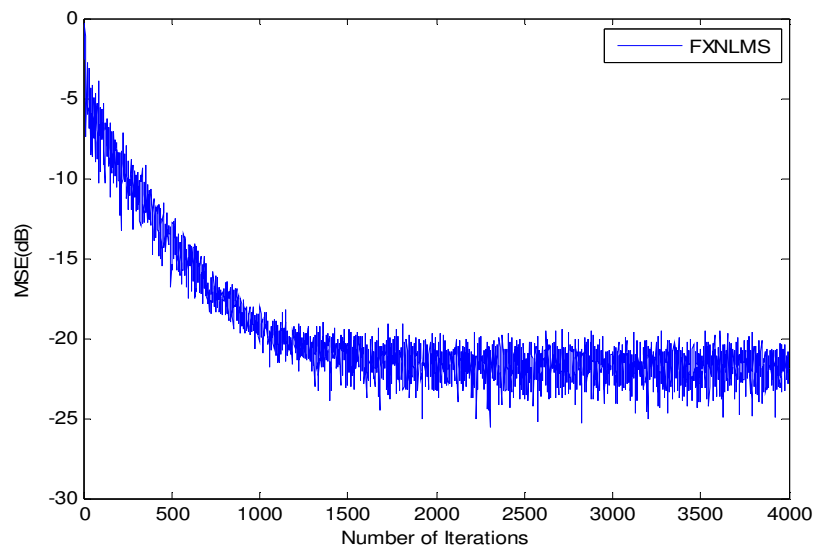


Fig. 5.13 MSE(dB) plot of FXNLMS algorithm for multichannel ANC.

5.10.2 Experiment II

In this experiment, experiment I is repeated considering logistic chaotic noise as the reference noise. The logistic chaotic noise is generated using the recursive equation [62], [64]

$$x(n+1) = \lambda x(n) [1 - x(n)] \quad (5.56)$$

where $x(0) = 0.9$, and $\lambda = 4$.

This nonlinear noise process is then normalized to have unity signal power. MSE(dB) is obtained for the proposed frequency domain multichannel FBFXNLMS algorithm and time domain FXNLMS algorithm. The step sizes used for FBFXNLMS and FXNLMS algorithm are $\mu = 0.09$ and $\mu = 0.07$ respectively. Simulation results for FBFXNLMS and FXNLMS algorithm are plotted in fig.5.12 and fig.5.13 respectively. Steady state MSE(dB) achieved by FBFXNLMS algorithm and FXNLMS algorithm is found to be same -21dB. From the results, it is evident that the proposed algorithm offers identical performance as the time domain FXNLMS algorithm for multichannel ANC.

5.10.3 Experiment III

In this experiment, a nonlinear multichannel ANC is considered. The ANC has one reference microphone, two secondary loudspeakers and four error microphones ($1 \times 2 \times 4$). The 2nd order Legendre neural network is employed for nonlinear ANC and is updated by the proposed frequency domain block Legendre FELMS (FBLFELMS) algorithm. Memory size N is chosen to be 10. The reference noise is taken as logistic chaotic noise defined in experiment II. This nonlinear noise process is then normalized to have unity signal power. In the experiment four linear primary path transfer functions considered are described below [62]

$$A_{11}(z) = [z^{-5} - 0.3z^{-6} + 0.2z^{-7}]$$

$$A_{12}(z) = [z^{-5} - 0.2z^{-6} + 0.1z^{-7}]$$

$$A_{13}(z) = [z^{-5} - 0.3z^{-6} + 0.1z^{-7}]$$

$$A_{14}(z) = [z^{-5} - 0.2z^{-6} + 0.2z^{-7}]$$

The eight secondary path transfer functions are non-minimum-phase FIR models and are described by the following equations.

$$B_{11}(z) = [z^{-2} + 1.5z^{-3} - z^{-4}]$$

$$B_{21}(z) = [z^{-2} + 1.7z^{-3} - z^{-4}]$$

$$B_{31}(z) = [z^{-2} + 1.8z^{-3} - z^{-4}]$$

$$B_{41}(z) = [z^{-2} + 1.9z^{-3} - z^{-4}]$$

$$B_{12}(z) = [z^{-2} + 1.5z^{-3} - z^{-4}]$$

$$B_{22}(z) = [z^{-2} + 1.2z^{-3} - z^{-4}]$$

$$B_{32}(z) = [z^{-2} + 1.1z^{-3} - z^{-4}]$$

$$B_{42}(z) = [z^{-2} + 1.0z^{-3} - z^{-4}]$$

Step size for FBLFELMS algorithm taken is $\mu=0.0009$. MSE(dB) obtained at each of the four error microphones and the overall MSE(dB) that combine results of the four error microphones are plotted in fig.5.14 , fig.5.15 , fig.5.16 , fig. 5.17, fig.5.18 respectively. For comparison with corresponding time domain algorithm combined MSE(dB) using time domain LFELMS algorithm is plotted in fig. 5.19. The step size and memory length for LFELMS algorithm are 0.0005 and 10 respectively. Steady state MSE(dB) achieved at all the error microphones is found to be same -25dB. From fig. 5.17 and fig.5.18 it is observed that overall MSE(dB) of FBLFELMS algorithm and LFELMS algorithm is -25dB. From the results, it is evident that the proposed algorithm offers identical performance as the time domain LFELMS algorithm for multichannel ANC but the real advantage of the proposed algorithm is large saving in computational complexity.

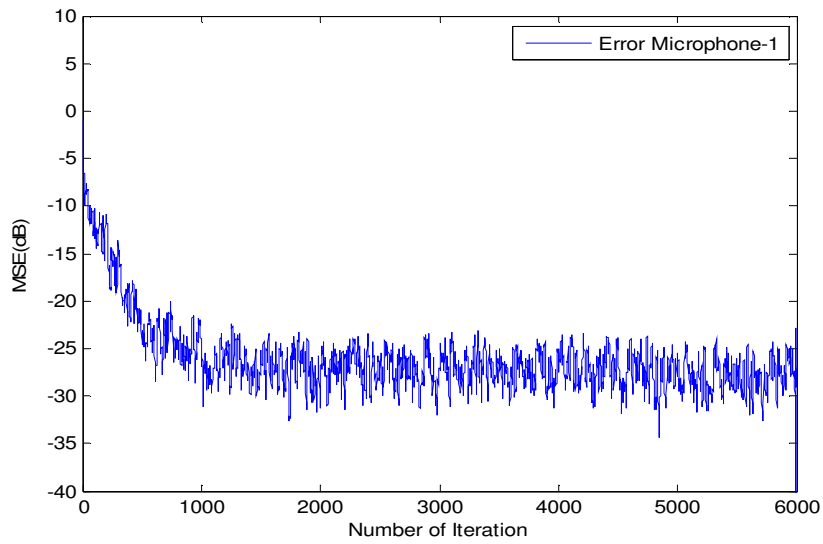


Fig.5.14 MSE(dB) plot of FBLFELMS algorithm at error microphone-1.

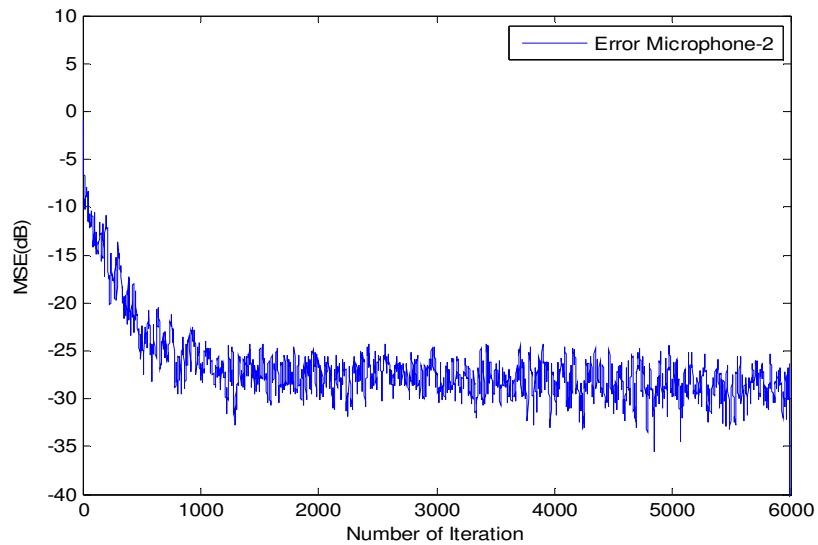


Fig.5.15 MSE(dB) plot of FBLFELMS algorithm at error microphone-2.

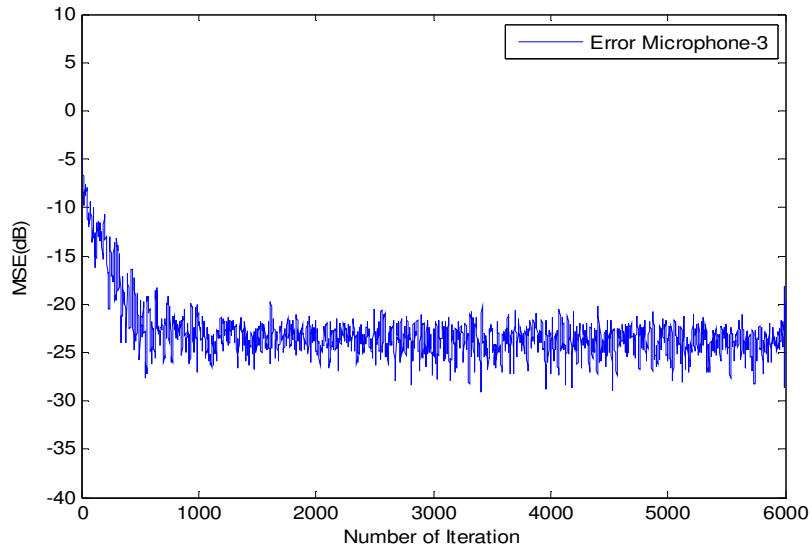


Fig.5.16 MSE(dB) plot of FBLFELMS algorithm at error microphone-3.

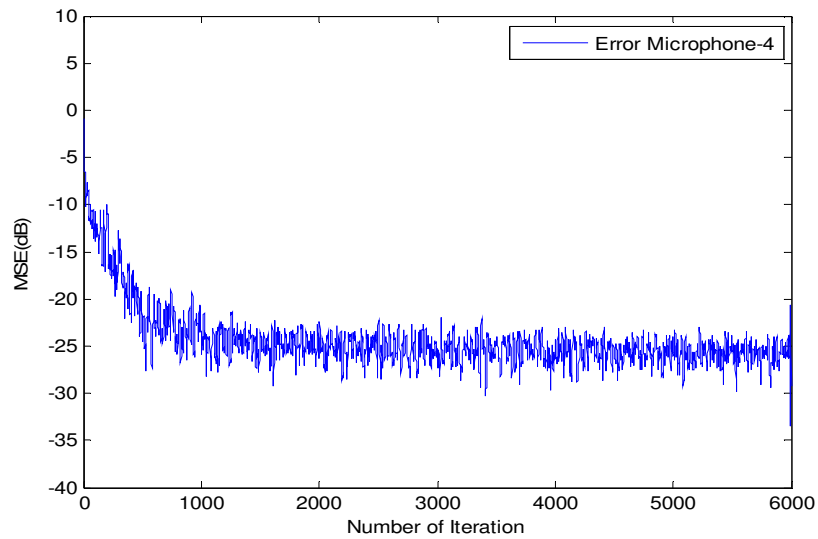


Fig.5.17 MSE(dB) plot of FBLFELMS algorithm at error microphone-4.

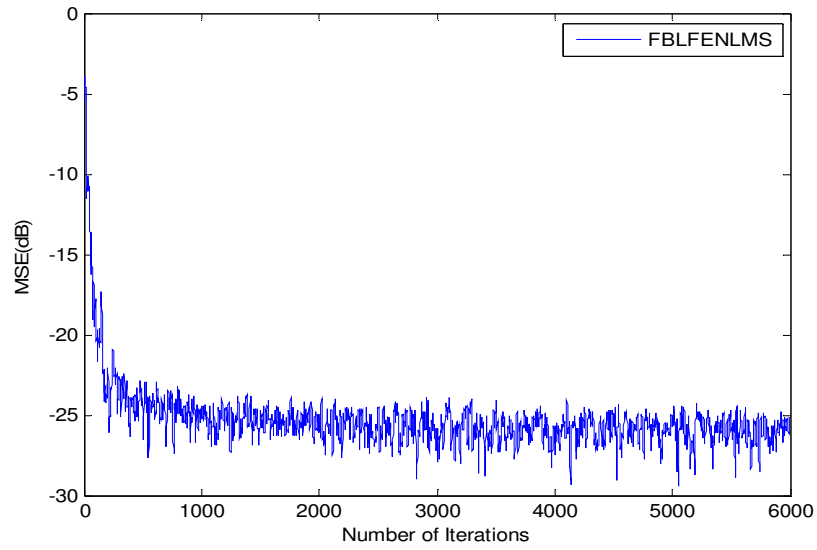


Fig.5.18 MSE(dB) plot of FBLFELMS algorithm of all the error microphones (combined).

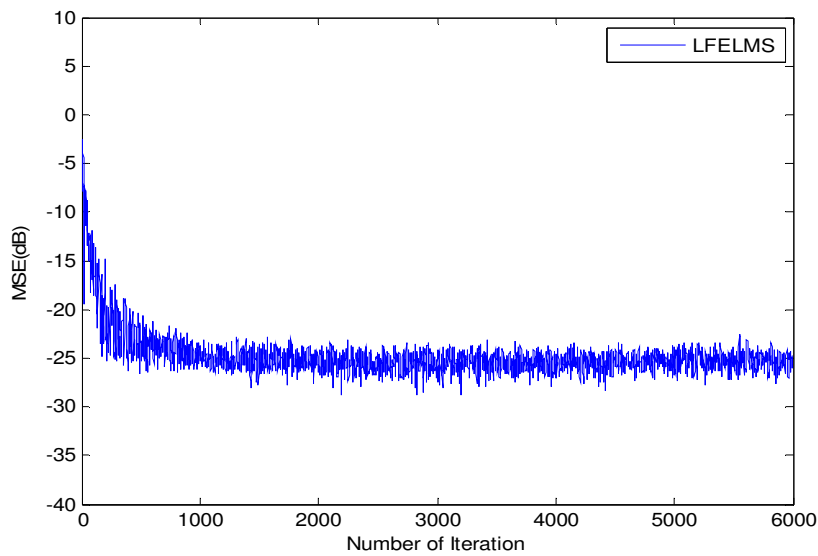


Fig.5.19 MSE(dB) plot of LFELMS algorithm of all the error microphones (combined).

5.10.4 Experiment on Real Time Signal

In an attempt to assess the performance of the proposed algorithms on signals picked up from real life environments, experiments are conducted once again. The reference signals considered are buccaneer jet cockpit noise, factory floor noise and white noise [79].

5.10.5 Experiment IV (Real Time Signal)

This experiment is conducted on three reference noise signals collected from real life environment. The setup is same as experiment I and the signals used are buccaneer jet cockpit noise, factory floor noise and white noise. The MSE(dB) plots of FXNLMS algorithm and FBFXNLMS algorithm for buccaneer jet cockpit noise, factory floor noise and white noise are shown in fig. 5.20 - fig. 5.25. The step sizes used for FBFXNLMS and FXNLMS algorithm for white noise are 0.09 and 0.07 respectively. Similarly step sizes for factory floor noise and Buccaneer jet cockpit noise are 0.06, 0.05 and 0.06, 0.05 respectively. From the results, it is evident that the proposed algorithm offers identical performance as the time domain FXNLMS algorithm for multichannel ANC.

5.10.6 Experiment V (Real Time Signal)

Experiment-III is repeated here considering the Buccaneer jet cockpit noise as reference signal. MSE(dB) for FBLFELMS algorithm obtained at each of the four error microphones and the overall MSE(dB) that combine results of the four error microphones are plotted in fig.5.26 - fig.5.30 respectively. Step size for FBLFELMS algorithm taken is $\mu=0.0009$. From the results, it is evident that the proposed algorithm offers good performance as for multichannel ANC but the real advantage of the proposed algorithm is large saving in computational complexity.

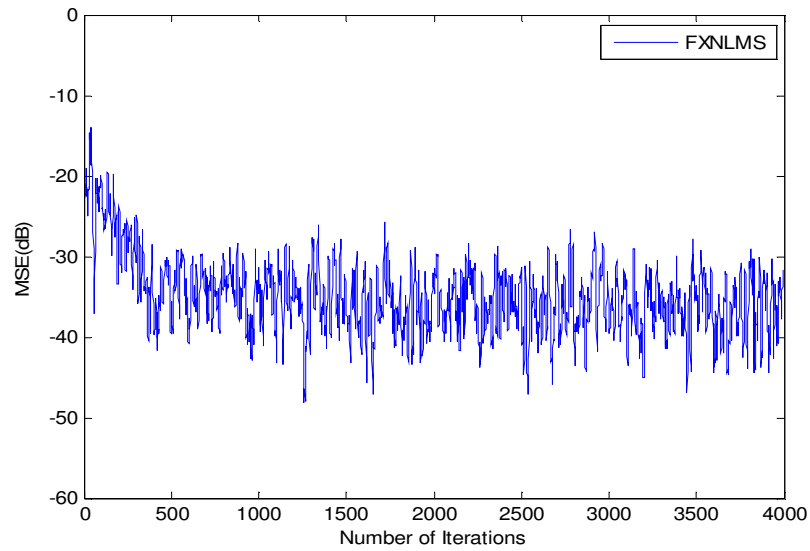


Fig. 5.20 MSE(dB) plot of FXNLMS algorithm for white noise.

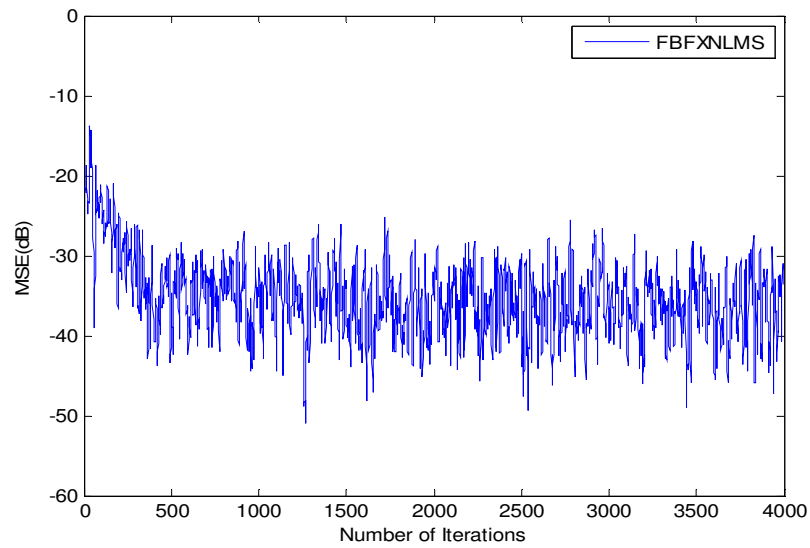


Fig. 5.21 MSE(dB) plot of FBFXNLMS algorithm for white noise.

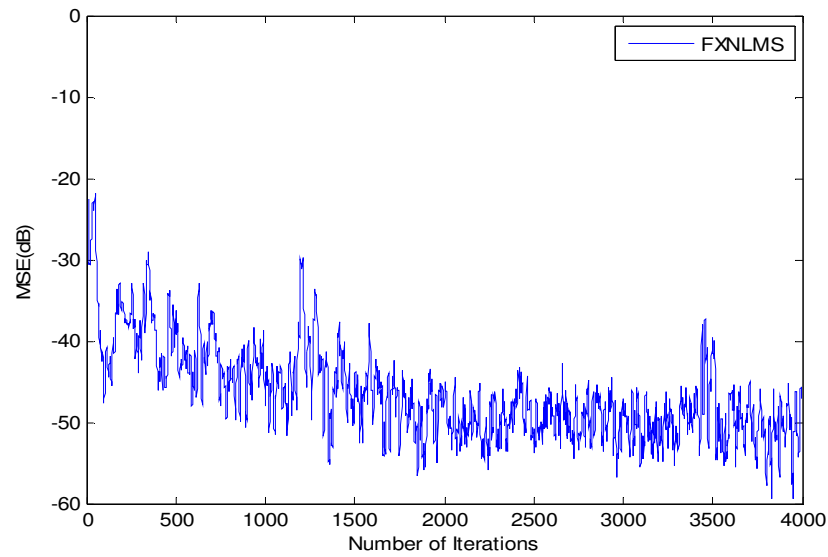


Fig. 5.22 MSE(dB) plot of FXNLMS algorithm for factory floor noise.

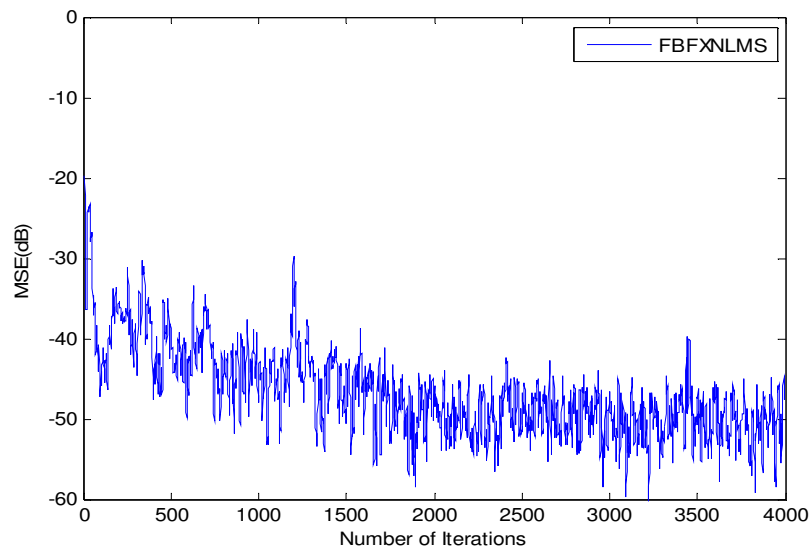


Fig. 5.23 MSE(dB) plot of FBFXLMS algorithm for factory floor noise.

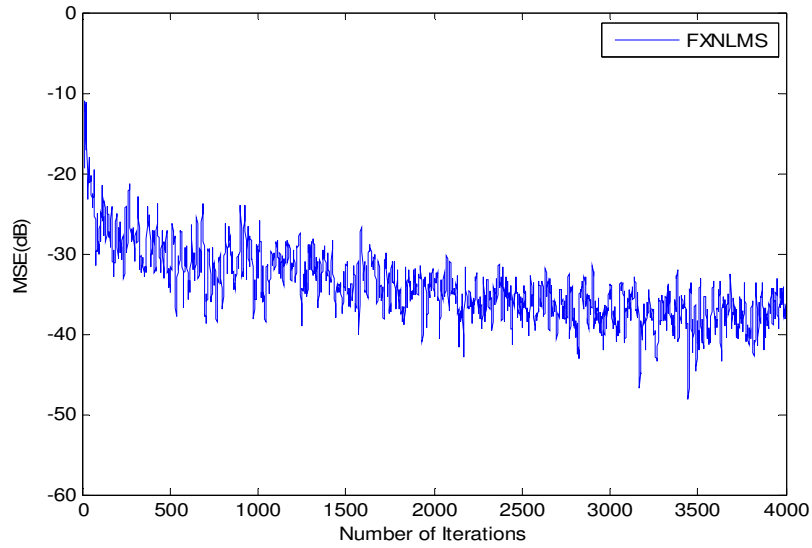


Fig. 5.24 MSE(dB) plot of FXNLMS algorithm for Buccaneer jet cockpit noise.

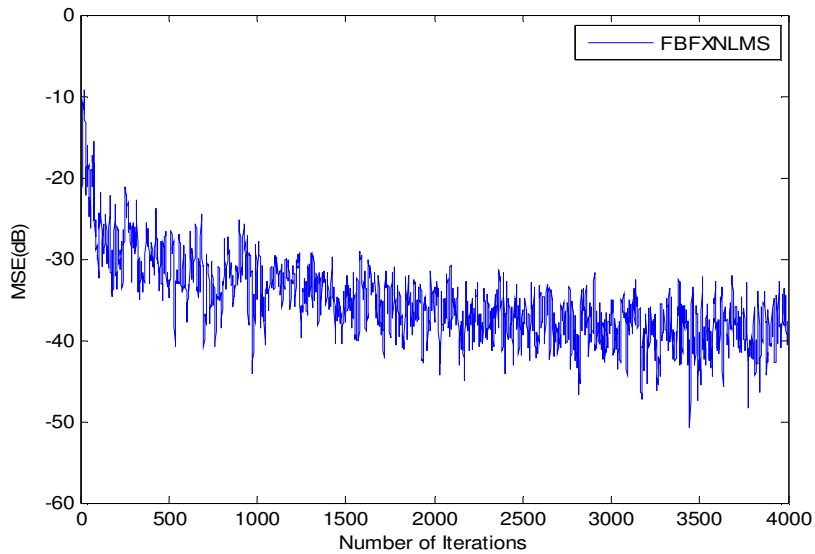


Fig. 5.25 MSE(dB) plot of FBFXNLMS algorithm for Buccaneer jet cockpit noise.

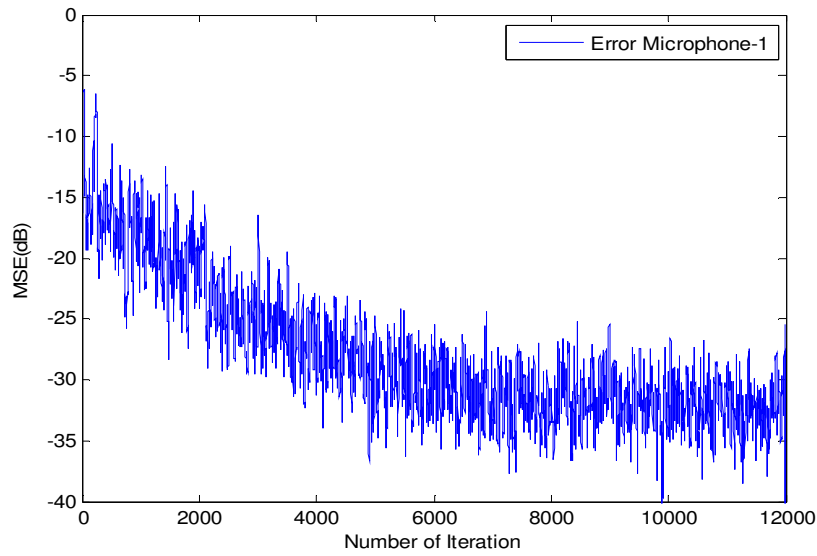


Fig.5.26 MSE(dB) plot of FBLFELMS algorithm for Buccaneer cockpit noise at error microphone-1.

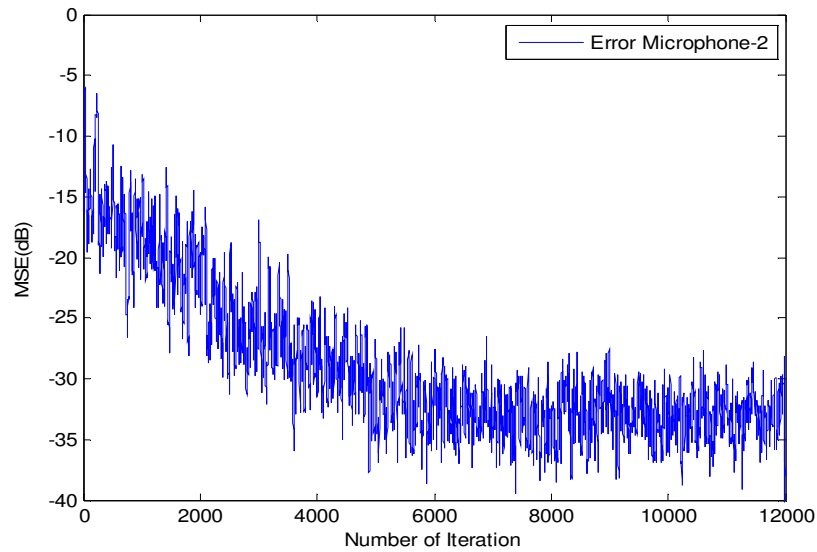


Fig.5.27 MSE(dB) plot of FBLFELMS algorithm for Buccaneer cockpit noise at error microphone-2.

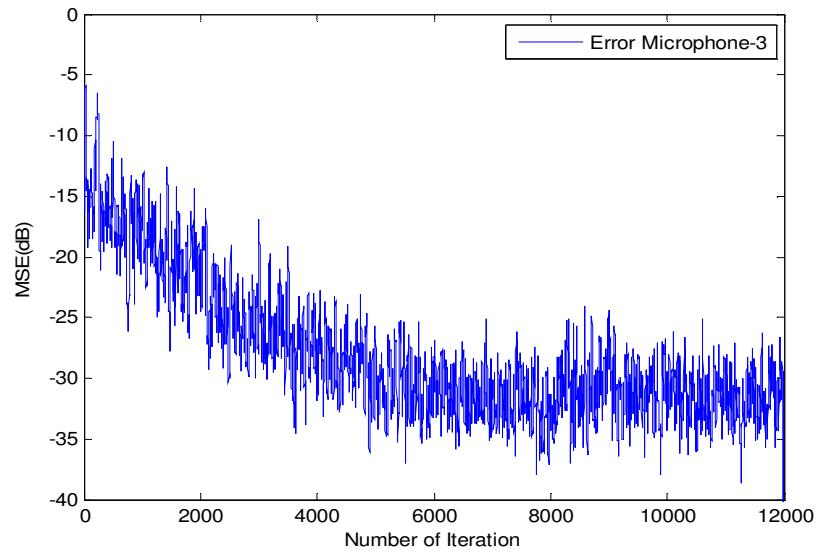


Fig.5.28 MSE(dB) plot of FBLFELMS algorithm for Buccaneer cockpit noise at error microphone-3.

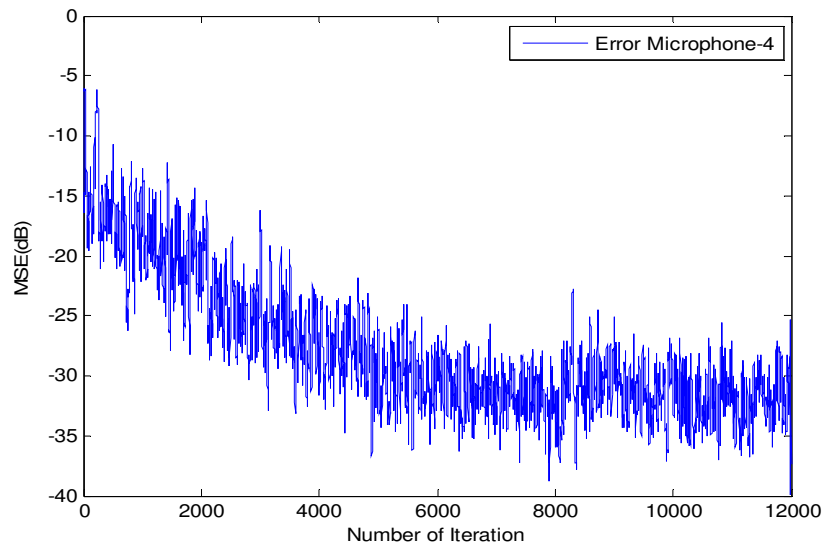


Fig.5.29 MSE(dB) plot of FBLFELMS algorithm for Buccaneer cockpit noise at error microphone-4.

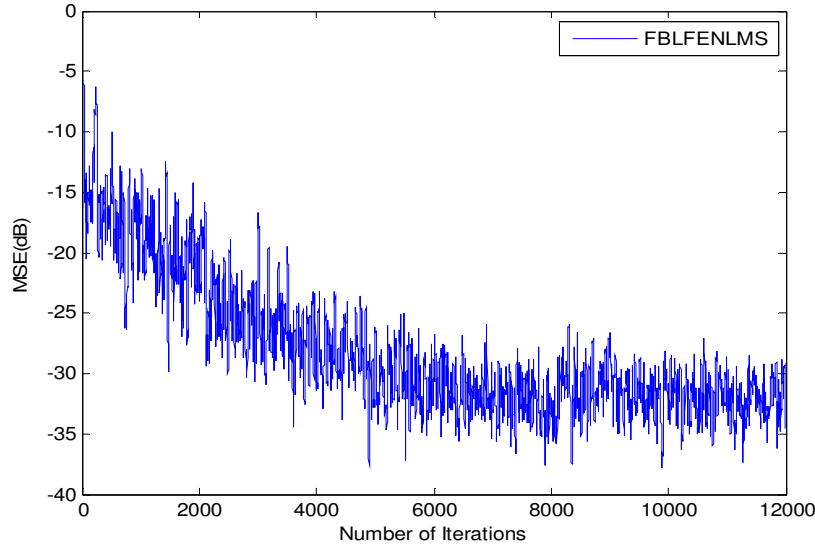


Fig.5.30 MSE(dB) plot of FBLFELMS algorithm of all the error microphones (combined).

5.11 Summary

This chapter deals with frequency domain implementation of multichannel ANC. Frequency domain implementation is made possible by block processing and using FFTs. Frequency domain block filtered-x normalized LMS (FBFXNLMS) algorithm is developed for noise mitigation in multichannel ANC. The proposed algorithm employed normalized LMS algorithm to facilitate variable step size control. Normally computational complexity requirement for frequency domain implementation is lower than its time domain counterpart. Here also computational complexity analysis of the developed algorithm is found to be much lower than time domain FXNLMS algorithm for multichannel ANC. Frequency domain block filtered-e LMS (FBFELMS) algorithm is developed which implement FELMS algorithm in frequency domain. New algorithms were developed for nonlinear ANC using Legendre neural network in the previous chapter. In this chapter frequency domain implementation of Legendre neural network for nonlinear ANC is also carried out. Detailed mathematical formulation of the algorithm for filter bank implementation is presented. Frequency domain Legendre filtered-e LMS

(FBLFELMS) is also developed which deals with frequency domain implementation of LFELMS algorithm. The validity of the proposed algorithms is demonstrated through extensive computer simulations. Performance of the proposed algorithms is also evaluated on signals collected from real life environment such as Bunnaneer jet cockpit noise, white noise and factory floor noise. The performance of proposed frequency domain algorithms is found to be equivalent to their time domain counterparts but the real advantage is in huge computational complexity reduction capability.

Chapter 6

Conclusion and Scope for Further Research

6.1 Conclusion

The work described in the thesis is primarily concerned with the study and development of novel nonlinear active noise control systems. The main focus of this research work is to design novel ANC based on conventional feedforward neural network topology. In particular, development of weight update algorithms for the proposed nonlinear ANC is the main challenge of this dissertation. Further, certain modifications in the existing and proposed algorithms have been incorporated to make them suitable for multichannel ANC. Development of adaptive algorithms for multilayer artificial neural network for nonlinear ANC with nonlinear secondary path model is one of the principal contribution of this dissertation. Development of adaptive algorithms for Legendre neural network in filter bank implementation is another major contribution of this thesis work. This dissertation work has also contributed immensely to the issues of frequency domain implementation of nonlinear ANC. Some of the major achievements of the present study are mentioned below.

One of the focus areas was on developing a MLP based neural network controller for nonlinear ANC. Two separate adaptive algorithms were developed for nonlinear ANC considering two situations (when secondary path was assumed linear or nonlinear). For nonlinear secondary path, adaptive algorithm was modified using virtual secondary path filter concept. By computer simulations it was observed that the proposed algorithms (using various nonlinear activation functions) outperformed standard VFXLMS algorithm in terms of steady state MSE(dB) with faster speed of convergence. In order to take advantage of low computational complexity of filtered error LMS algorithm, both the developed algorithms were suitably modified to develop NFELMS algorithm. Performance of the modified algorithms were also analyzed by computer simulation on real life signals and compared with that of VFELMS algorithm. Finally it has been concluded that the proposed adaptive algorithms for MLP based ANC controller are superior to the standard Volterra based algorithms.

Another primary area of focus was to explore Legendre neural network for nonlinear ANC. The adaptive algorithm for Legendre neural network used as controller of ANC was first developed. The algorithm was found to be simple and easy to implement and has low computational complexity. In order to reduce computational complexity further a reduced structure Legendre neural network was proposed. The reduced structure Legendre neural network with its reduced computational complexity was found out to be performing well with reference to steady state MSE(dB) level. LFXLMS algorithm, LFELMS algorithm and LFXRLS algorithm were developed for reduced structure LNN based on FXLMS algorithm, FELMS algorithm and FXRLS algorithm respectively. The developed algorithms require less computation compared to FFXLMS algorithm. The developed algorithms were modified to deal with nonlinear secondary path (NSP) which relies upon virtual secondary path concept. In order to reduce computational complexity faster version of LFXLMS algorithm was also developed.

The third zone of focus has been frequency domain implementation of multichannel ANC. Conventionally computational complexity requirement of frequency domain implementation is lower than its time domain counterpart. So FBFXNLMS algorithm was developed for multichannel ANC using efficient DSP tools like FFT and IFFT. The developed algorithm uses NLMS algorithm to facilitate variable step size. Observing that FELMS algorithm is an efficient alternative to FXLMS algorithm with an advantage of reduction in computational complexity frequency domain block filtered-e least mean square (FBFELMS) algorithm was proposed. The reduced structure Legendre neural network for nonlinear ANC was also implemented in frequency domain. Two new adaptive algorithms, frequency domain block Legendre filtered-x least mean square (FBLFXLMS) algorithm and frequency domain block Legendre filtered-e least mean square (FBLFELMS) algorithm were developed. Analysis of the results of computer simulation on synthetic data and real life data revealed identical performance (MSE(dB) and speed of convergence) with reference to the proposed frequency domain algorithms and their corresponding time domain algorithms but with lower computational complexity.

6.2 Scope for Further Research

To conclude the thesis, following are some pointers for further work.

- Multichannel implementation of neural ANC could be a research area, which should be explored. Multichannel implementation requires simultaneous operation of many neural networks which increases the complexity of the system. Study can be done to effectively manage the networks to optimize the overall performance of the ANC.
- The implementation aspects in FPGA or hybrid FPGA and DSP combined platforms of some of these algorithms and structures should also be investigated. Implementing the whole system in FPGA or hybrid FPGA and DSP through many challenges.
- Focus be concentrated to design efficient filter structures based on new topologies to address the ANC problems.
- Different novel methods based on evolutionary and bio-inspired techniques be developed with the basic objective to optimally train the weights of the adaptive filter structures. Emergence of evolutionary and bio-inspired techniques and their highly effective variants can really change the world of ANC.

References

- [1] S. M. Kuo and D. R. Morgan, “Active Noise Control Systems: Algorithm and DSP Implementations,” New York, Wiley, 1996.
- [2] S. J. Elliott, “Signal Processing for Active Control,” London, Academic, 2001.
- [3] B. Widrow and S. D. Stearns, “Adaptive Signal Processing,” Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [4] H. Coanda, Brevet d’Invention No. 722.274, Republique Francaise (1930).
- [5] P. Lueg, “Process of Silencing Sound Oscillations,” U.S. Patent No. 2043416, June, 1936.
- [6] H. F. Olson and E. G. May, “Electronic Sound Absorber,” Journal of Acoustic society, AM, 25, pp. 1130-1136, November 1953.
- [7] S. Haykin, “Adaptive Filter Theory,” 3rd edition, Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [8] A. V. Oppenheim and R. W. Schaffer, “Digital Signal Processing”, Englewood Cliffs, NJ: Printice-Hall, 1975.
- [9] S. Haykin, “Neural Networks: A Comprehensive Foundation,” Englewood Cliffs, NJ: Macmillan, 1994.
- [10] S.M. Kuo and D. R. Morgan, “Active Noise Control: A Tutorial Review,” Proceedings of the IEEE, vol. 87, no. 6, 1999.
- [11] Paul Strauch, B. Mulgrew, “Active Control of Nonlinear Noise Processes in a Linear Duct,” IEEE Transaction Signal Processing, vol. 46, no. 9, pp. 2404-2412, September 1998.
- [12] Victor E. DeBrunner and Dayong Zhou, “Hybrid Filtered Error LMS Algorithm: Another Alternative to Filtered-x LMS,” IEEE Transaction on Circuits Systems I, Regular Papers, vol. 53, no. 3, pp. 653–661, March 2006.
- [13] Scott D. Snyder and Colin H. Hansen, “The Effect of Transfer Function Estimation Errors on the Filtered LMS Algorithm”, IEEE Transactions on Signal Processing, vol. 42, no. 4, April 1994.

- [14] E. A. Wan, “Adjoint-LMS: An Efficient Alternative to Filtered-x LMS and Multiple Error LMS Algorithms,” In Proceedings of ICASSP 1996, vol.3, pp.1842-1845, 1996.
- [15] S. R. Popovich, “Simplified Parameter Update for Identification of Multiple Input, Multiple Output Systems,” in Proceedings of Internoise, pp.1129–1232, 1994.
- [16] Debi Prasad Das, Danielle J. Moreau and Ben S. Cazzolato, “Adjoint Nonlinear Active Noise Control Algorithm for Virtual Microphone,” Journal on Mechanical Systems and Signal Processing, vol. 27, pp.743–754, 2012.
- [17] A. Roure and A. Albarrazin “The Remote Microphone Technique for Active Noise Control,” in Proceedings of Active 1999, pp. 1233–1244, Florida, USA, 2000.
- [18] Shiang-Hwua Yu and Jwu-Sheng Hu, “Controller Design for Active Noise Cancellation Headphones Using Experimental Raw Data,” IEEE/ASME Transactions on Mechatronics, vol. 6, no. 4, pp.483-490, December 2001
- [19] Ying Song, Yu Gong, and Sen M. Kuo, “A Robust Hybrid Feedback Active Noise Cancellation Headset,” IEEE Transactions on Speech and Audio Processing, vol. 13, no. 4, pp. 607-617, July 2005.
- [20] Thomas Schumacher, Hauke Krüger, Marco Jeub, Peter Vary, Christophe Beaugeant, “Active Noise Control in Headsets: A New Approach for Broadband Feedback ANC,” ICASSP, pp-417-420, 2011.
- [21] Cheng-Yuan Chang and Sheng-Ting Li, “Active Noise Control in Headsets by Using a Low-Cost Microcontroller,” IEEE Transactions on Industrial Electronics, vol. 58, no. 5, pp. 1936-1942, May 2011.
- [22] Romain Serizel, Marc Moonen, Jan Wouters, and Søren Holdt Jensen, “Integrated Active Noise Control and Noise Reduction in Hearing Aids,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, no. 6, pp. 1137-1146, August 2010.
- [23] Romain Serize, Marc Moonen, Jan Wouters and Søren Holdt Jensen, “A Zone of Quiet Based Approach to Integrated Active Noise Control and Noise Reduction in Hearing Aids,” IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp.229-232, October 18-21, New Paltz, NY, 2009.

- [24] W. S. Gan and S. Kuo, "An Integrated Audio and Active Noise Control Headsets," *IEEE Transactions on Consumer Electronics*, vol. 48, pp. 242–247, 2002.
- [25] Casper K. Chen, Tzi-Dar Chiueh and Jyh-Horng Chen, "Active Cancellation System of Acoustic Noise in MR Imaging," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 2, pp. 186-191, February 1999.
- [26] Kuan-Hung Cho, Tzi-Dar Chiueh, Ching-Po Lin, Casper K. Chen, Jyh-Horng Chen, "An Active Noise Cancellation System for fMRI," *Proceedings of NFSI & ICFBI*, Hangzhou, China, pp.265-276, October 12-14, 2007.
- [27] Govind Kannan, Ali A. Milani, Issa M.S. Panahi, Nasser Kehtarnavaz, "Performance Enhancement of Adaptive Active Noise Control Systems for fMRI Machines," *32nd Annual International Conference of the IEEE EMBS*, Buenos Aires, Argentina, pp.4327-4330, August 31 - September 4, 2010.
- [28] M. Kida, R. Hiramaya, Y. Kajikawa, T. Tani, Y. Kurumi, "Head-mounted Active Noise Control System for MR Noise," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp-245-248, April 2009.
- [29] Lichuan Liu, Shruthi Gujjula and Sen M. Kuo, "Multi-channel Real Time Active Noise Control System for Infant Incubators," *31st Annual International Conference of the IEEE EMBS Minneapolis, Minnesota, USA*, pp. 935-938, September 2-6, 2009.
- [30] X. Qiu and C. H. Hansen, "An Algorithm for Active Control of Transformer Noise with On-Line Cancellation Path Modelling Based on The Perturbation Method," *Journal of Sound and Vibration*, no. 240(4), pp.647-665, 2001.
- [31] Ji-guang Jiang, Deng-feng Wang, Yue Zeng, Jun-ting Wang, Xiao-lin Cao, "Active Control of Vehicle Interior Noise Based on Selective Attenuation Method," *Second International Conference on Information and Computing Science*, pp. 157-160, 2009.
- [32] V. J. Mathews, "Adaptive Polynomial Filters," *IEEE Signal Processing Magazine*, pp.10-26, July 1991.
- [33] V. J. Mathews and G. L. Sicuranza, "Polynomial Signal Processing," New York: John Wiley & Sons, 2000.

- [34] L. Tan and J. Jing, "Filtered-X Second Order Volterra Adaptive Algorithm," *Electronics letters*, vol. 33, no. 8, pp. 671-672, April 1997.
- [35] L. Tan and J. Jing, "Adaptive Volterra Filters for Active Control of Nonlinear Noise Processes," *IEEE Transactions on Signal Processing*, vol. 49, no.8, pp. 1667-1676 August 2001.
- [36] Debi Prasad Das, Swagat Ranjan Mohapatra, Aurobinda Routray and T. K. Basu, "Filtered-s LMS Algorithm for Multichannel Active Control of Nonlinear Noise Processes," *IEEE transactions on Speech and Audio Processing*, 2006.
- [37] S. M. Kuo and H.-T. Wu, "Active Noise Control Systems with Adaptive Nonlinear Filters," *Proceedings of the 2004 IEEE International Conference on Control Applications Taipei, Taiwan*, pp.1330-1335, September 2-4, 2004.
- [38] S. M. Kuo and H.-T. Wu, "Nonlinear Adaptive Bilinear Filters for Active Noise Control Systems," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. 52, no. 3, pp.617-624, March 2005.
- [39] R. T. Bammbang, L. Anggono and K. Uchida, "DSP Based RBF Neural Modeling and Control for Active Noise Cancellation," *Proceedings of the IEEE International Symposium on Intelligent Control, Vancouver, Canada*, pp. 460-466, October 2002.
- [40] R. T. Bambang, "Adjoint EKF Learning in Recurrent Neural Networks for Nonlinear Active Noise Control," *Applied Soft Computing*, vol.8, pp. 1498–1504, 2008.
- [41] R. T. Bambang, "Active Noise Cancellation Using Recurrent Radial Basis Function Neural Network," *Asia-Pacific Conference on Circuits and Systems*, vol. 2, pp. 231-236, 2002.
- [42] Jiang Lifei, "A Design of an Adaptive Active Noise Control System in Driver's cab," *The 2nd International Conference on Computer and Automation Engineering*, vol. 3, pp.560-563, 2010.
- [43] C.-Y. Chang and K.-K. Shyu, "Active Noise Cancellation with a Fuzzy Adaptive Filtered-X Algorithm," *IEE Proceeding Circuits Devices System*, vol. 150, no. 5, pp. 416-422, October 2003.

- [44] Cheng-Yuan Chang and Kuo-Kai Shyu, "A Self-Tuning Fuzzy Filtered-U Algorithm for the Application of Active Noise Cancellation," *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, vol. 49, no. 9, pp. 1325-1333, September 2002.
- [45] H. V. Tuan and D. H. Nghia, "A Fuzzy Neural Network Feedback Active Noise Controller," *10th International Conference on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, pp.1109-1114, 17–20 December 2008.
- [46] J. Liu, J. Sun and G. Wei, "Application of an ANFIS Algorithm to Narrowband Active Noise Control System," *9th International Conference on Signal Processing*, pp. 284-287, 26-29 October 2008.
- [47] N. Azadi, A. Ohadi, "Enhanced Multi-Channel Active Fuzzy Neural Network Noise Control in an Enclosure," *Proceedings of the World Congress on Engineering*, vol. III, London, U.K, July 6 - 8, 2011.
- [48] F. Russo and G. L. Sicuranza, "Genetic Optimization in Nonlinear Systems for Active Noise Control: Accuracy and Performance Evaluation," *International Conference on Instrumentation and Measurement Technology Conference*, Sorrento, Italy, pp. 1512-1517, 24-27 April 2006.
- [49] F. Russo and G. L. Sicuranza, "Accuracy and Performance Evaluation in the Genetic Optimization of Nonlinear Systems for Active Noise Control," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 7, pp. 1443-1450, August 2007.
- [50] C.-Y. Chang and D.-R. Chen, "Active Noise Cancellation Without Secondary Path Identification by Using an Adaptive Genetic Algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 9, pp. 2315-2327, September 2010.
- [51] N. K. Rout, D. P. Das and G. Panda, "Particle Swarm Optimization Based Active Noise Control Algorithm Without Secondary Path Identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 2, pp. 554-563, February 2012.

- [52] S. D. Snyder and N. Tanaka, “Active Control of Vibration using Neural Networks,” *IEEE Transactions on neural Networks*, vol.6, no.4, pp. 819-828, July 1995.
- [53] M. Bouchard, B. Paillard and C.T. Dinh, “Improved Training of Neural Network for the Nonlinear Active Control of Sound and Vibration,” *IEEE Transactions on neural Networks*, vol.10, no.2, pp. 391-401, March 1999.
- [54] M. Bouchard, “New Recursive -Least-Squares Algorithms for Nonlinear Active Control of Sound and Vibration Using Neural Networks,” *IEEE Transactions on neural Networks*, vol.12, No.1, pp. 135-147, January 2001.
- [55] M. O. Tokhi and R. Wood, “Active Noise Control Using Radial Basis Function Networks,” *Control Eng. Practice*, vol. 5, No. 9, pp.1311-1322, 1997.
- [56] C. Y. Chang and F. B. Luoh, “Enhancement of Active Noise Control Using Neural Based Filtered-x Algorithm,” *Journal of Sound and Vibration*, vol. 305, pp. 348-356, August 2007.
- [57] Xinghua Zhang, Xuemei Ren, JingNa, BoZhang and HongHuang, “Adaptive Nonlinear Neuro-Controller with an Integrated Evaluation Algorithm for Nonlinear Active Noise Systems,” *Journal of Sound and Vibration*, vol. 329, pp. 5005–5016, 2010.
- [58] Cheng-Yuan Chang, “Neural Filtered-u Algorithm for the Application of Active Noise Control System with Correction Terms Momentum,” *Digital Signal Processing a Review Journal*, vol. 20, pp. 1019-1026, 2010.
- [59] Kunal Kumar Das and Jitendriya Kumar Satapathy, “New Neural Network Algorithms for Nonlinear Active Noise Cancellation with Nonlinear Secondary Path,” *IEEE International Conference on Signal Processing, Communication, Computing and Networking Technologies*, Kumaracoil, Thuckalay, Tamilnadu, India, pp. 286-290, July 2011.
- [60] C, Y. Chang and S. T. Pan, “Neural Based Active Noise Controller,” *Proceedings Sixth International Conference on Machine learning and Cybernatics*, pp.425-430, Hong Kong, August 2007.

- [61] M. Ayala Botto, J.M.C. Sousa and J.M.G. Sa da Costa, “ Intelligent Active Noise Control Applied to a Laboratory Railway Coach Model,” *Control Engineering Practice*, vol. 13, pp. 473–484, 2005.
- [62] D. Zhou and V. DeBrunner, “ Efficient Adaptive Nonlinear Filters for Nonlinear Active Noise Control,” *IEEE Transactions on Circuit and Systems-I*, vol. 54, no. 3, pp. 669–681, March 2007.
- [63] Y. H. Pao “Adaptive Pattern Recognition and Neural Networks”, Addison-Wesley, Reading, MA, 1989.
- [64] D. P. Das and G. Panda, “ Active Mitigation of Nonlinear Noise Processes Using a Novel Filtered-S LMS Algorithm,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 3, pp. 313–322, May 2004.
- [65] G. L. Sicuranza and A. Carini, “A Generalized FLANN Filter for Nonlinear Active Noise Control,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2412–2417, November 2011.
- [66] D. S. Nelson, S. C. Douglas, and M. Bodson, “Fast Exact Adaptive Algorithms for Feedforward Active Noise Control,” *Proceedings of International Journal Adaptive Control Signal Processing*, vol. 14, no. 6, pp. 643–661, 2000.
- [67] E.P. Reddy, D. P. Das and K. M. M. Prabhu, “Fast Exact Multichannel FSLMS Algorithm for Active Noise Control,” *Journal of Signal Processing*, vol. 89, pp. 952–956, 2009.
- [68] E. R. Ferrara, “Fast Implementation of LMS Adaptive Filters,” *IEEE Transaction on Acoustic, Speech and Signal Processing*, vol. 28, pp. 474–475, August 1980.
- [69] J. Benesty and P. Duhamel “A Fast Exact Least Mean Square Adaptive Algorithm”, *IEEE Transaction on Signal Processing*, vol. 40, no. 12, pp. 2904–2912, December 1992.
- [70] S. C. Douglas “An Efficient Implementation of the Modified Filtered-X LMS algorithm”, *IEEE Signal Processing Letters*, vol. 4, no. 10, pp. 286–288, October 1997.

- [71] S. C. Douglas, "Fast Implementations of the Filtered-X LMS and LMS Algorithms for Multichannel Active Noise Control," *IEEE Transaction on Speech and Audio Processing*, vol.7, pp. 454-465, July 1999.
- [72] B. Farhang-Boroujeny, "Fast LMS/Newton Algorithms Based on Autoregressive Modeling and their Applications to Acoustic Echo Cancellation," *IEEE Transaction on Signal Processing*, vol.45, pp. 1987-2000, August 1997.
- [73] Q. Shen and A.S. Spanias "Time and Frequency Domain X Block LMS Algorithms for Single Channel Active Noise Control," *Proceeding 2nd Congress on Recent Development in Air and Structure Borne Sound and Vibration*, Auburn, AL,USA, pp. 353-360, March.
- [74] G. A. Clark, S. K. Mitra and S. R. Parkar, "Block Implementation of Adaptive Digital Filters," *IEEE Transaction Circuits and Systems*, vol CAS-28, pp 584-592, June 1981 and *IEEE Transaction Acoustic Speech, Signal Processing*, Joint Special Issue on Adaptive Signal Processing, ASSP-29, pp. 744-752, June 1981.
- [75] S. M. Kuo, M. Tahernezehadi and L. Ji, "Frequency-Domain Periodic Active Noise Control and Equalization," *IEEE Transactions on Speech and Audio Processing* vol. 5, no. 4, PP. 348-358, July 1997.
- [76] M. R. Asharif, T. Takebayashi, T. Chugo and K. Murano, "Frequency-domain Noise Canceller: Frequency-bin Adaptive Filtering (FBAF)," in *Proceedings ICASSP*, Tokyo, Japan, pp. 41.22.1-4, April 1986.
- [77] K. Reichard and D. Swanson, "Frequency-domain Implementation of the Filtered-x Algorithm with Online System Identification," *Proceeding 2nd Conference on Recent Development in Active control of Sound and Vibration*, Blacksburg, VA, pp. 562-573, April 1993.
- [78] D. P. Das, G. Panda and S. M. Kuo, "New Block filtered -x LMS Algorithm for Active Noise Control Systems," *IET Signal Processing*, vol. 1(2), pp. 73-81, 2007.
- [79] M. Salmasi, H. Mahdavi-Nasab and H. Pourghassem, "Comparison of Feed-Forward and Recurrent Neural Networks in Active Cancellation of Sound Noise," *International Conference on Multimedia and Signal Processing*, pp.25-29, 2011.

Publication

- [P1] Kunal Kumar Das and Jitendriya Kumar Satapathy, “New Neural Network Algorithms for Nonlinear Active Noise Cancellation with Nonlinear Secondary Path,” IEEE International Conference on Signal Processing, Communication, Computing and Networking Technologies, Kumaracoil, Thuckalay, Tamilnadu, India, pp. 286-290, July 2011.
- [P2] Kunal Kumar Das and Jitendriya Kumar Satapathy, “AdjointLMS Algorithm Based on Functional Expansion for Nonlinear Active Noise Cancellation,” IEEE National Conference on Computational Intelligence, Control, and Computer Vision in Robotics & Automation, Rourkela, pp.213-217, March 2008.
- [P3] Kunal Kumar Das and Jitendriya Kumar Satapathy, “Novel Algorithms Based on Legendre Neural Network for Nonlinear Active Noise Control with Nonlinear Secondary Path,” International Journal of Computer Science and Information Technologies, vol. 3(5), pp. 5036 - 5039, 2012.
- [P4] Kunal Kumar Das and Jitendriya Kumar Satapathy, “Frequency-Domain Block Filtered-x NLMS Algorithm for Multichannel ANC,” IEEE First International Conference on Emerging Trends in Engineering and Technology, Nagpur, pp.1293-1297, July 2008.
- [P5] Kunal Kumar Das and Jitendriya Kumar Satapathy, “Legendre Neural Network for Nonlinear Active Noise Cancellation with Nonlinear Secondary Path,” IEEE International Conference on Multimedia Signal Processing and Communication Technologies, Aligarh, India, pp.40-43, December 2011.

BIO-DATA OF THE CANDIDATE

Name of the candidate : Kunal Kumar Das.

Father's Name : Prasanta Kumar Das.

Date of Birth : 10th September 1975.

Permanent Home Address : AT-Purunahatsahi,
(Near-Binod Bhaban)
PO-Baripada,
Dist-Mayurbhanj,
ORISSA-757001.

Phone : +91-9437218383.

E-Mail : kunalbapu@gmail.com

ACADEMIC QUALIFICATION :

2012 : PhD dissertation submitted, NIT Rourkela.

2001 : M.E. in Electronic Systems & Communication Engineering, REC, Rourkela.

1997 : B.E. in Electrical Engineering, UCE, Burla.

PROFESSIONAL EXPERIENCE : 14 years of teaching and research experience.